

Discovering Beaten Paths in Collaborative Ontology-Engineering Projects using Markov Chains

Simon Walk^{a,*}, Philipp Singer^b, Markus Strohmaier^{b,c}, Tania Tudorache^d, Mark A. Musen^d, Natalya F. Noy^d

^a*Institute for Information Systems and Computer Media, Graz University of Technology, Austria*

^b*GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany*

^c*Dept. of Computer Science, University of Koblenz-Landau, Germany*

^d*Stanford Center for Biomedical Informatics Research, Stanford University, USA*

Abstract

Biomedical taxonomies, thesauri and ontologies in the form of the International Classification of Diseases as a taxonomy or the National Cancer Institute Thesaurus as an OWL-based ontology, play a critical role in acquiring, representing and processing information about human health. With increasing adoption and relevance, biomedical ontologies have also significantly increased in size. For example, the 11th revision of the International Classification of Diseases, which is currently under active development by the World Health Organization contains nearly 50,000 classes representing a vast variety of different diseases and causes of death. This evolution in terms of size was accompanied by an evolution in the way ontologies are engineered. Because no single individual has the expertise to develop such large-scale ontologies, ontology-engineering projects have evolved from small-scale efforts involving just a few domain experts to large-scale projects that require effective collaboration between dozens or even hundreds of experts, practitioners and other stakeholders. Understanding the way these different stakeholders collaborate will enable us to improve editing environments that support such collaborations. In this paper, we uncover how large ontology-engineering projects, such as the International Classification of Diseases in its 11th revision, unfold by analyzing usage logs of five different biomedical ontology-engineering projects of varying sizes and scopes using Markov chains. We discover intriguing interaction patterns (e.g., which properties users frequently change after specific given ones) that suggest that large collaborative ontology-engineering projects are governed by a few general principles that determine and drive development. From our analysis, we identify commonalities and differences between different projects that have implications for project managers, ontology editors, developers and contributors working on collaborative ontology-engineering projects and tools in the biomedical domain.

Keywords: Collaborative ontology engineering; Markov chains; sequential patterns; collaboration; ontology-engineering tool; user interface

1. Introduction

Today, biomedical ontologies play a critical role in acquiring, representing and processing information about human health. For example, the International Classification of Diseases (ICD) is a taxonomy that is used in more than 100 countries to encode patient diseases, to compile health-related statistics and to collect health-related spending statistics. Similarly, the National Cancer Institute's Thesaurus (NCIt) represents an important OWL-based vocabulary for classifying cancer and cancer-related terms.

With their increase in relevance, biomedical taxonomies, thesauri and ontologies have also significantly increased in size to cover new findings and to extend and complement their original areas of application. For example, the 11th revision of the International Classification of Diseases (ICD-11), currently under active development by the World Health Organization

(WHO), consists of nearly 50,000 classes representing a vast variety of different diseases and causes of death. In contrast to previous revisions, the foundation component of ICD-11 is implemented as an OWL ontology with a broader scope than previous ICD revisions.

This growth was accompanied by a need to adapt the way these ontologies are engineered as no single individual or small group of domain experts have the expertise to develop such large-scale ontologies. New tools and processes have to be developed in order to coordinate, augment and manage collaboration between the dozens or hundreds of experts, practitioners and stakeholders when engineering an ontology.

Understanding the ways in which such a large number of participants – e.g., more than 100 experts contribute to ICD-11 – collaborate with one another when creating a structured knowledge representation is a prerequisite for quality control and effective tool support.

Objectives: Consequently, we aim at understanding how large collaborative ontology-engineering projects such as ICD-

*Corresponding author (simon.walk@tugraz.at)

11 unfold. In particular, we want to investigate if we can identify usage patterns in the change-logs of collaborative ontology-engineering projects? We approach this problem by analyzing patterns in usage logs of five biomedical ontology-engineering projects of varying sizes and scopes. For this analysis we employ Markov chain models for investigating and modeling sequential interaction paths (c.f. Section 3.2). Such paths are represented by chronologically ordered lists of interactions within the underlying ontology for (a) a single user or (b) a single class (see Figure 2). For example, we study sequences of properties that were either changed by (a) *a single user* on any class or (b) *a single class* by any user in an ontology over time. For example, as depicted in Figure 2, a sequential property path for a single user (user-based) consists of a chronologically ordered list of all properties (e.g., *title*, *definition* etc.), which have been changed by that user on any class, while a sequential property path for a single class (class-based) consists of a chronologically ordered list of properties that were changed on that class by any user. Instead of only modeling sequences for single users or classes, our data contains a set of paths; e.g., each path in the dataset consists of sequences of properties whose value has been changed by a single user over time. This allows us to tap into accumulated patterns. Concretely, we are interested in studying emerging patterns of subsequent steps in such sequential paths – e.g., which properties do users frequently change after a specific given property.

The analyzed datasets range from large-scale datasets such as ICD-11 to smaller ones such as the Ontology for Parasite Lifecycle (OPL). Given the differences of our datasets in a number of salient characteristics, we investigate if specific patterns can be found across all or only in certain biomedical ontology-engineering projects. Furthermore, we investigate and discuss features of these projects that potentially affect observed patterns, which can only be found in specific datasets. This analysis can be seen as a stepping stone for collaborative ontology-engineering project managers to devise infrastructures and tool support to augment collaborative ontology engineering.

Contributions: We present new insights on social interactions and editing patterns that suggest that large collaborative ontology-engineering projects are governed by a few general principles that determine and drive development. Specifically, our results indicate that general edit patterns can be found in all investigated datasets, even though they (i) represent different projects with different goals, (ii) use variations of the same ontology-editors and tools for the engineering process and (iii) differ in the way the projects are coordinated.

To the best of our knowledge, the work presented in this paper represents the most fine-grained and comprehensive study of patterns in large-scale collaborative ontology-engineering projects in the domain of biomedicine. In addition, our analysis is conducted across five datasets of different sizes, which have been developed using different versions of Collaborative Protégé (Table 1).

2. Collaborative ontology engineering

According to Gruber [1], Borst [2], Studer et al. [3] an ontology is an explicit specification of a shared conceptualization. In particular, this definition refers to a machine-readable construct (the formalization) that represents an abstraction of the real world (the shared conceptualization), which is especially important in the field of computer science as it allows a computer (among other things) to “understand” relationships between entities and objects that are modeled in an ontology.

Collaborative ontology engineering is a new field of research with many new problems, risks and challenges that we must first identify and then address. In general, contributors of collaborative ontology-engineering projects, similar to traditional collaborative online production systems¹ (e.g., Wikipedia), engage remotely (e.g., via the internet or a client-server architecture) in the development process to create and maintain an ontology. As an ontology represents a formalized and abstract representation of a specific domain, disagreements between authors on certain subjects can occur. Similar to face-to-face meetings, these collaborative ontology-engineering projects need tools that augment collaboration and help contributors in reaching consensus when modeling topics of the real world.

Indeed, the majority of the literature about collaborative ontology engineering sets its focus on surveying, finding and defining requirements for the tools used in these projects [4, 5].

The Semantic Web community has developed a number of tools aimed at supporting the collaborative development of ontologies. For example, Semantic MediaWikis [6] and its derivatives [7, 8, 9] add semantic, ontology modeling and collaborative features to traditional MediaWiki systems.

Protégé, and its extensions for collaborative development, such as WebProtégé and iCAT [10] (see Figure 1 for a screenshot of the iCAT ontology-editor interface) are prominent stand-alone tools that are used by a large community worldwide to develop ontologies in a variety of different projects. Both WebProtégé and Collaborative Protégé provide a robust and scalable environment for collaboration and are used in several large-scale projects, including the development of ICD-11 [11].

Pöschko et al. [12] and Walk et al. [13] have created *PragmatiX*, a tool to visualize and analyze a collaboratively engineered ontology and aspects of its history and the engineering process, providing quantitative insights into the ongoing collaborative development processes.

Falconer et al. [14] investigated the change-logs of collaborative ontology-engineering projects, showing that users exhibit specific roles, which can be used to group and classify users, when contributing to the ontology. Pesquita and Couto [15] investigated whether the location and specific structural features can be used to determine if and where the next change is going to occur in the Gene Ontology².

¹Note that the term traditional online production systems refers to online platforms that have users collaborate in engineering digital goods, opposed to a structured knowledge base that is the result of collaborative ontology-engineering.

²<http://www.geneontology.org>

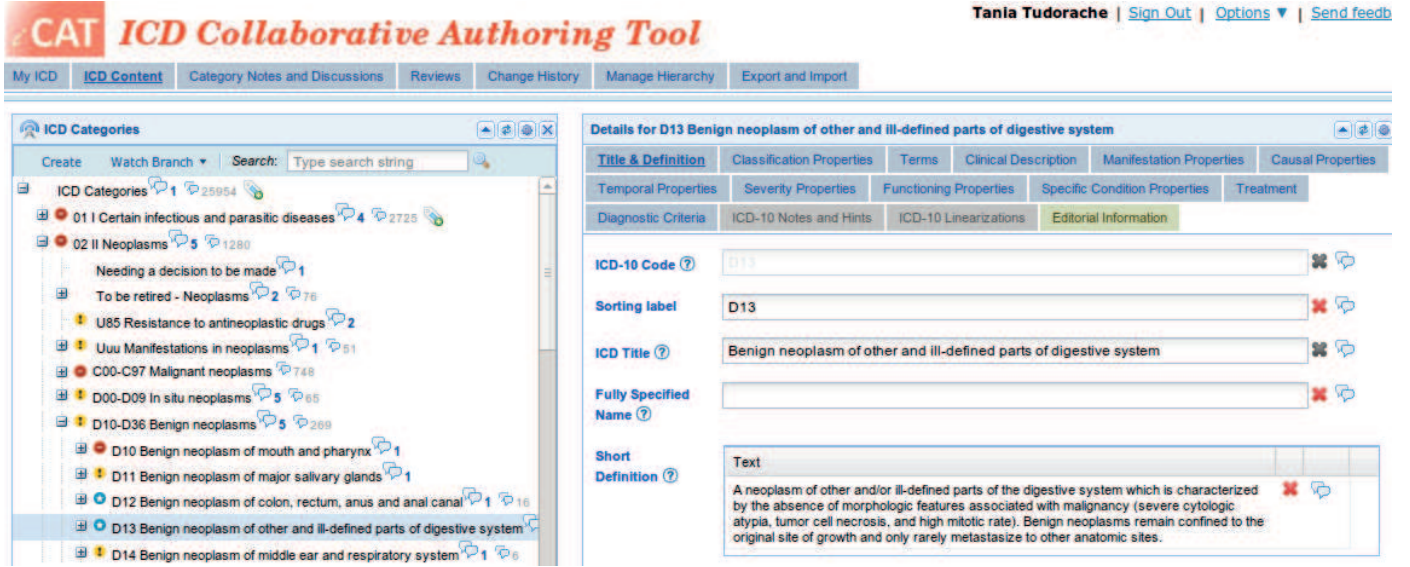


Figure 1: A screenshot of iCAT, a custom tailored, web-based version of WebProtégé, developed for the collaborative engineering of ICD-11. The left part of the interface visualizes the ICD-11 class hierarchy, the class titles, the number of annotations each class has received (speech bubbles) and its overall progress (color and symbol before the class title). The right part of the interface shows the different user-interface sections (e.g. *Title & Definition* or *Classification Properties*), listing all properties and property values for each class.

Goncalves et. al [16, 17, 18] performed an analysis of different versions of ontologies by applying and categorizing *Diff* algorithms, with the goal of categorizing the differences between consecutive and chronologically ordered versions of the ontologies. Furthermore, they conducted reasoner performance tests and identified factors that potentially increase reasoner performance. For the analysis presented in this paper we were able to rely on ChAO [19], which is a change-log provided by Protégé and its derivatives that already provides us with detailed and unambiguous logs of changes for the investigated ontologies.

In a similar context Grau et al. [20, 21] proposed a logical framework for modularity of ontologies and a definition of what is to be considered as an ontology module. In general, an ontology module can be used to extract the meaning of a specified set of terms from an ontology. Extracting the right amount of information is especially important for the topic of ontology reuse. According to Grau et al. modularity also represents a crucial factor in collaborative ontology-engineering environments as modular representations of ontologies are easier to understand, to extend and to reuse, similar to modularity in software engineering projects.

Mikroyannidi et al. [22] investigated the detection and use of (design) patterns in the content of an ontology, using a clustering approach. In contrast to Mikroyannidi et al., our analysis focuses on the detection of sequential patterns in interaction data rather than content.

Strohmaier et al. [23] investigated the hidden social dynamics that take place in collaborative ontology-engineering projects from the biomedical domain and provides new metrics to quantify various aspects of the collaborative engineering processes. Wang et al. [24] have used association-rule mining to analyze user editing patterns in collaborative ontology-engineering projects. The approach presented in this paper uses

Markov chains to extract much more fine grained user-interaction patterns incorporating a variable number of historic editing information.

The only requirement to perform the pattern analysis that we present in this paper is the availability of a structured log of changes that can be mapped to the underlying ontology. The majority of the discussed collaborative ontology-engineering environments provide such a log, allowing for a similar analysis. For example, the Semantic MediaWikis store all the changes to the articles, and thus the ontology, allowing to expand the application of Markov chains to analyze sequential patterns as shown in this paper.

3. Materials & methods

For the analysis conducted in this paper we concentrated our efforts on five ontology-engineering projects in the biomedical domain. Each of the projects (i) has at least two users who contributed to the project, (ii) provides a structured log of changes and (iii) represents knowledge from the biomedical domain. In Section 3.1 we provide a brief history for each dataset and in Section 3.2 we describe the sequential path analysis. To aid readers in understanding the analyses conducted in this paper and its implications we provide a very brief overview of Markov chains and the involved model selection methodology in Section 3.3.

3.1. Datasets

Table 1 lists the detailed features and observation periods for the following five datasets that we used in our analysis. All datasets have been created either with WebProtégé or special

Table 1: Detailed information of the datasets used for the sequential pattern analysis to extract beaten paths in collaborative ontology-engineering projects.

Ontology	classes changes DL expressivity	ICD-11	ICTM	NCIt	BRO	OPL
		48,771 439,229 <i>SHOIN(D)</i>	1,506 67,522 <i>SHOIN(D)</i>	102,865 294,471 <i>SH</i>	528 2,507 <i>SHIF(D)</i>	393 1,993 <i>SHOIF</i>
Editor	tool	iCAT	iCAT-TM	Collaborative Protégé	WebProtégé	Collaborative Protégé
Users	users	109	27	17	5	3
	bots (changes)	1 (935)	1 (1)	0 (0)	0 (0)	0 (0)
Duration	first change	18.11.2009	02.02.2011	01.06.2010	12.02.2010	09.06.2011
	last change	29.08.2013	17.7.2013	19.08.2013	06.03.2010	23.09.2011
	observation period (ca.)	4 years	2.5 years	3 years	1 month	3 months

versions of WebProtégé. To be able to conduct the pattern detection analysis for a different dataset, there is only one requirement that needs to be satisfied: The availability of a change-log that can be mapped onto the ontology so that changes can be associated with users and classes without ambiguity.

The DL expressivity [25, 26] of the five datasets is added to Table 1 to highlight that the investigated ontologies exhibit different strategies regarding their OWL-DL expressivity. As all levels of expressivity shown in Table 1 allow for the definition and assignment of properties and classes, they do not influence the conducted pattern detection analyses. Also, in the case of WebProtégé and its derivatives, the data used for the pattern detection analysis can be extracted from the change-logs, allowing us to prevent parsing and extracting values from OWL directly.

The International Classification of Diseases (ICD)³ is the international standard for diagnostic classification used to encode information relevant to epidemiology, health management, and clinical use in over 100 United Nations countries. The World Health Organization (WHO) develops ICD, and publishes new revisions of the classification every decade or more. The current revision in use is ICD-10, a taxonomy that contains over 15,000 classes. The 11th revision of ICD,⁴ **ICD-11**, is currently taking place and brings two major changes with respect to previous revisions. First, ICD-11’s foundation component is developed as an OWL ontology using a much richer representation formalism than previous revisions. ICD-11 contains very detailed descriptions of several aspects of diseases, mostly represented as properties in the ontology. Second, the development of ICD-11 takes place in a Web-based collaborative environment, called iCAT (see Figure 1), which allows domain experts around the world to contribute and review the ontology online. ICD-11 is planned to be finalized in May 2017.

The International Classification of Traditional Medicine (ICTM) is a WHO led project that aimed to produce an international standard terminology and classification for diagnoses and interventions in Traditional Medicine.⁵ ICTM, similarly to ICD-11, implements an OWL based ontology as foundation component, which tries to unify the knowledge from the traditional medicine practices from China, Japan and Korea. Its content is authored in 4 languages: English, Chinese, Japanese and Korean. More than 20 domain experts from the three countries

developed ICTM using a customized version of the iCAT system, called iCAT-TM. The development of ICTM was stopped in 2012, and a subset of ICTM is also included as a branch in the ICD-11 ontology.⁶

The National Cancer Institute’s Thesaurus (NCIt) [27] has over 100,000 classes and has been in development for more than a decade. It is a reference vocabulary covering areas for clinical care, translational, basic research, and cancer biology. A multidisciplinary team of editors works to edit and update the terminology based on their respective areas of expertise, following a well-defined workflow. A lead editor reviews all changes made by the editors. The lead editor accepts or rejects the changes and publishes a new version of the NCI Thesaurus. The NCI Thesaurus is, at its core, an OWL ontology, which uses many OWL primitives such as defined classes and restrictions. It was named thesaurus due to historical reasons, however fully conforms to OWL semantics, thus represents an actual ontology.

The Biomedical Resource Ontology (BRO) originated in the Biositemaps project,⁷ an initiative of the Biositemaps Working Group of the NIH National Centers for Biomedical Computing [28]. Biositemaps is a mechanism for researchers working in biomedicine to publish metadata about biomedical data, tools, and services. Applications can then aggregate this information for tasks such as semantic search. BRO is the enabling technology used in Biositemaps; a controlled terminology for describing the resource types, areas of research, and activity of a biomedical related resource. BRO was developed by a small group of editors, who use a Web-based interface (WebProtégé) to modify the ontology and to carry out discussions to reach consensus on their modeling choices.

The Ontology for Parasite Lifecycle (OPL) models the life cycle of the *T.cruzi*, a protozoan parasite, which is responsible for a number of human diseases. OPL is an OWL ontology that extends several other OWL ontologies. It uses many OWL constructs such as restrictions and defined classes. Several users from different institutions collaborate on OPL development. This ontology is much smaller and has far fewer users than NCIt, ICD-11, or ICTM.

⁶The ICD-11 dataset used in our analysis did not include the ICTM branch.

⁷<http://biositemaps.ncbcs.org>

³<http://www.who.int/classifications/icd/en/>

⁴<http://www.who.int/classifications/icd/ICDRevision/>

⁵<http://tinyurl.com/ictmbulletin>

3.2. Sequential interaction paths

For our sequential pattern analysis we analyze three different kinds of paths, which all represent interactions with the underlying ontology. A sequential path is represented by the chronologically ordered list of extracted interactions for either a single user or a single class (see Figure 2). For example, a sequential property path for a single user (user-based) consists of a chronologically ordered list of all properties (e.g., *title*, *definition* etc.), which have been changed by that user on any class, while a sequential property path for a single class (class-based) consists of a chronologically ordered list of properties that were changed on that class by any user.

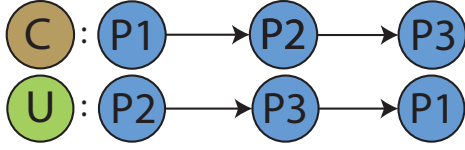


Figure 2: The top row of the figure depicts an exemplary **class-based** sequential property path (P1 to P3) for class C. This means that for class C the property P1 was changed first, then property P2 and most recently changed was property P3. The bottom row of the figure depicts the sequential property path (P1 to P3), however this time for a user U (**user-based**). Analogously, user U has first changed P2, continued to change property P3 and most recently changed P1.

User-sequence paths: First, we analyze activity patterns within the collaborative ontology-engineering project. This means that we analyze sequences of users who change a class. We want to detect and describe the different sequential patterns (the structure) that can be extracted from the change-logs of the investigated collaborative ontology-engineering projects.

Structural paths: Analogously to the User-Sequence Paths, we investigate edit-strategies, such as *bottom-up* or *top-down* development, that users follow. Is it possible to detect common patterns of which depth level a user frequently contributes to after a given current depth level? In addition to development-strategies, we look at the relationships (e.g., parent, child, sibling, etc.) between the current and the next class a user is going to contribute to.

Property paths: On a content-based level, we investigate the series of property-changes users perform on. In particular, we want to identify common successive property-changes – i.e., which properties *users* (user-based) regularly change consecutively and which properties are changed back-to-back for *classes* (class-based).

3.3. Markov chain models

For the analysis conducted in this paper we are adopting the methodology presented by Singer et al. [29] and mapped to collaborative ontology-engineering change logs by Walk et al. [30] to detect sequential patterns identified in and extracted from change-logs of collaborative ontology-engineering projects.

For a better understanding of the collected results, we will provide a short description of Markov chains. For an in-depth description of our methodology we point to Singer et al. [29], Walk et al. [30].

In general, Markov chain models are used for stochastically modeling transitions between states on a given state space. In our case, a Markov chain consists of a finite *state-space* (e.g., properties that a user edits over time; see Section 3.2) and the corresponding *transition probabilities* (e.g., the probability of changing property j after property i) between these states. Markov chain models are usually described as memoryless which means that the next state in a sequence only depends on the current one and not on a sequence of preceding ones (also known as Markovian property). Hence, this property defines serial dependence between adjacent nodes in trajectories – this is where the term “chain” comes from. Such a model is usually called a *first-order* or *memoryless* model.

As we are interested in modeling sequential interaction paths of collaborative ontology-engineering projects (see Section 3.2), we fit a Markov chain model on such sequences $D = (x_1, x_2, \dots, x_n)$ with states from a finite set S . Then, we can write the Markovian property as:

$$P(x_{n+1}|x_1, x_2, \dots, x_n) = P(x_{n+1}|x_n) \quad (1)$$

After the model fitting on the data, a Markov chain model is usually represented via a stochastic transition matrix P with elements $p_{ij} = P(x_j|x_i)$ where it holds that for all i :

$$\sum_j p_{ij} = 1 \quad (2)$$

For our analysis, we will make use of these transition probabilities to identify likely transitions for a variety of different states.⁸ For example, if we fit the Markov chain model on sequential property paths for users (see Section 3.2), element p_{ij} of the transition matrix would tell us the probability that users change property j right after i (e.g., in 60% of all cases). By now, e.g., looking for the highest transition probabilities from state i to all other states of S , we can identify potential high-frequent patterns in our data.

4. Results

4.1. User-sequence paths

In the *User-Sequence Paths* analysis we investigate patterns emerging when looking at sequences of users who contribute to a class of an ontology. Hence, given a sequence of n contributors for a class over time, we identify consecutive users who edit the class (e.g., user Y frequently contribute to a class after user X).

Analyzing the chronologically ordered list of contributors for each class of the five investigated datasets provides the necessary information to identify users who perform changes on classes after (or before) other users. Note that this analysis on its own, without regarding additional factors, such as the

⁸Note that throughout this article we usually refer to the entities modeled (i.e., interactions) instead of states. However, we speak about transition probabilities between these entities as we derive them directly from the resulting model transition matrix.

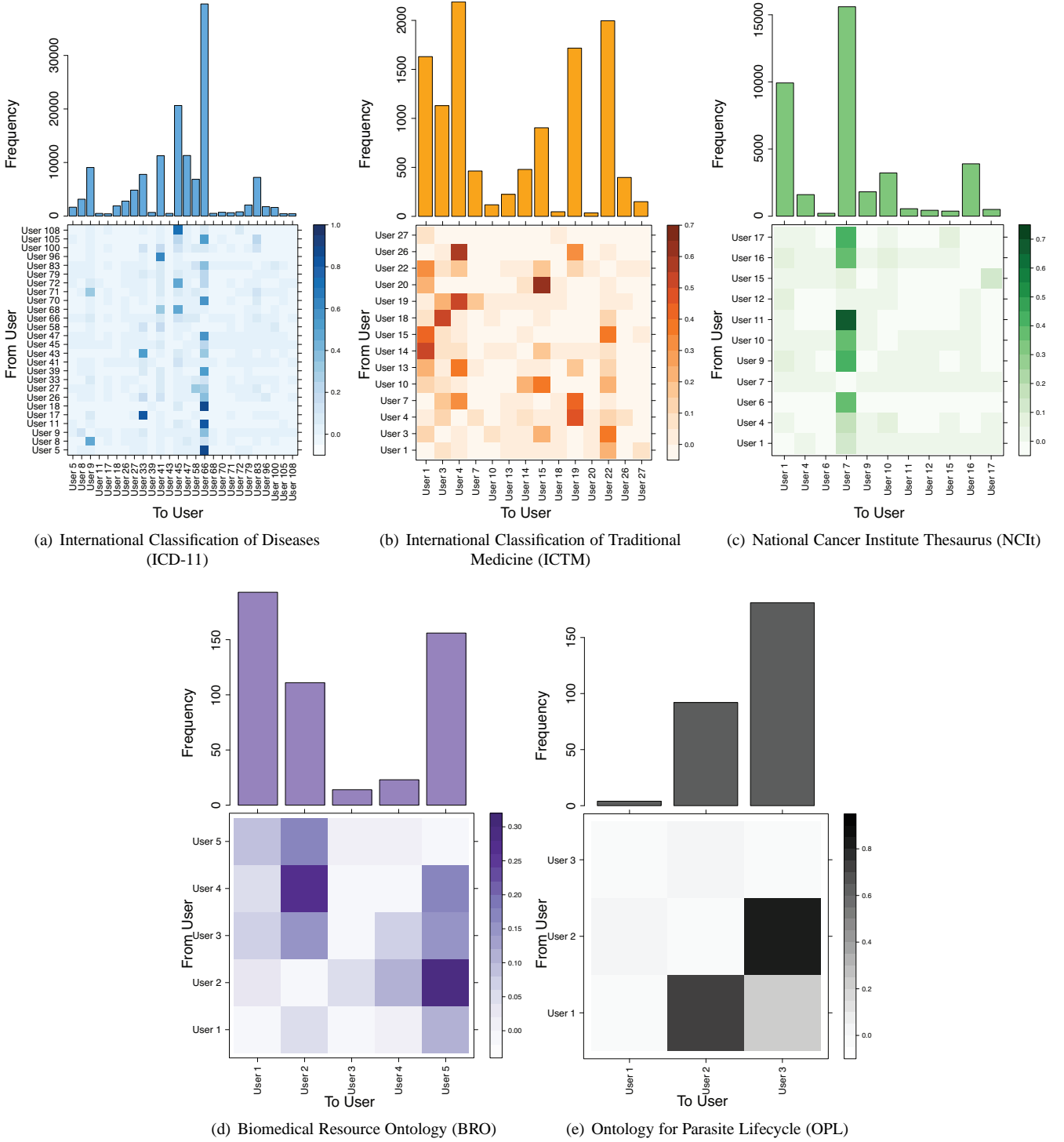


Figure 3: Results for the User-Sequence Paths analysis: The columns and rows of the transition maps (**bottom area** of Figures 3(a) to 3(e)) represent the transition-probabilities between the users of each dataset for a first-order Markov chain, where rows are *source users* and columns are *target users*. A sequence (or transition-probability) is always read *from row to column*. Darker colors represent higher transition-probabilities while lighter colors indicate lesser transition-probabilities. Absolute probability values are dependent on the number of investigated rows and columns, hence relative differences are of greater importance. Darker colored columns identify gardeners, a contributor focused on pruning ontology classes and fixing syntactical errors. The histograms (**top area** of Figures 3(a) to 3(e)) show the number of changes performed by each user (again for a first-order Markov chain) within the five ontologies in alphabetical order. Note, that the y-axes for all histograms are scaled differently for each dataset. All datasets have a few users who contributed the majority of changes, while the rest of the users (the long-tail) only contributed a very small number of changes. Note that the transition-probabilities depicted in the transition maps are relative numbers for each column and row individually. The sum of all transition probabilities for one row in the transition maps is 1. For example, if *User 1* exhibits a transition probability of 0.30 to another *User 2* it means that *User 2* has a 30% probability of changing a class after *User 1*. Thus, an inspection of the transition maps **and** histograms is necessary for proper interpretation. To increase readability we have removed users from the plots who have contributed only a very limited number of changes for ICD-11, ICTM and NCIt.

changed property or the performed change-action, does not provide information about actual collaboration. The results of this analysis could be used to potentially identify users who work on the same classes, however, we do not know if they actually collaborate with or just clean up (i.e., a *gardener*, a contributor focused on pruning ontology classes and fixing syntactical errors) after other users.

Path & model description: To analyze user sequences, we iterated over each class of our datasets and extracted a chronologically ordered list of contributors. For example, a given path for a given class can look like the following: *User A, User B, User B, User C*. As we are interested in uncovering patterns of distinct users, we merged multiple consecutive changes by the same user into a single change – our previous example would then unfold into: *User A, User B, User C*. By doing so we remove biases emerging when one single user consecutively changes the same class over and over as this may result in unreasonable high transition probabilities between equal users.

We fit a first-order Markov chain model on this set of paths, where each path represents a single class of the ontology and each element of a path constitutes a change by a single user on the class. The resulting transition probabilities between users then e.g., tell us the probability that *User B* changed a class after *User A*. Hence, they give us thorough insights into frequent consecutive user patterns that emerge when looking at which users contribute to classes in an ontology. Due to reasons of privacy we obfuscated the usernames and replaced them with generic names.

Results: When investigating the transition probabilities (representing a Markov chain of first order) between contributors (see bottom area of Figures 3(a) to 3(e)) we can identify very active users by looking at darker colored columns of the transition maps. Note that these darker colored columns can also be used to identify gardeners, a contributor focused on pruning ontology classes and fixing syntactical errors. As we have merged all consecutive changes of the same user into one single change, the diagonal, representing the transition probabilities between the same users, is 0. The absolute transition probabilities, depicted next to each transition map, are dependent on the absolute amount of observations and users, thus are to be interpreted relatively to each other for each row individually. When looking at the probabilities between the three most active users (being users 66, 45 and 47), and all corresponding target users in ICD-11 we can see that the probabilities are very evenly distributed among them. Meaning that, when investigating the rows (*From User*) that correspond to the top three most active users, probabilities to all target users (*To User*) are very evenly distributed, with very minor exceptions. This indicates that users who contribute many changes to ICD-11 are not followed by specific other contributors, but exhibit an even distribution of users that edited a class after them. Nonetheless, we can clearly identify *User 66* to be the most likely user that edits a class after nearly all other users. This suggests, that *User 66* may represent a gardener, a contributor focused on pruning ontology classes and fixing syntactical errors, in ICD-11.

For NCIt we can clearly observe that *User 7* appears to be a *gardener*, who is checking all the changes contributed by all

other users. For BRO *Users 2* and *5* are prominent target users, evident in the high transition probabilities as *To User* (dark columns) – i.e., they frequently edit a class after other users do. Interestingly, the user with the highest number of changes (*User 1*) exhibits very low and evenly distributed transition probabilities (row) and is not necessarily the user that most likely changes a class after another users. This shows us that there does not need to be a necessary connection between the overall activity of users and their activity as a gardener. This could also mean that *User 1* is possibly working independently from the other users in BRO, or that *User 1* is a domain specialist and all other users only change concepts that have not been worked on by that specialist. However, further investigations in future work are required to confirm this observation as our Markov chain analysis is not able to determine this kind of distinction. For OPL we can observe that *User 3* frequently changes the same classes after *User 2*. A similar observation can be made for *Users 1* and *2*. However, one has to keep in mind that *User 1* has contributed a limited number of changes, rendering the observed transition probabilities less useful as they rarely occur.

The histograms (see top area of Figures 3(a) to 3(e)) indicate that a small number of users contribute the majority of changes (similar to a long-tail distribution). However, this appears to be more dominant for specific ontologies compared to others. In order to measure the inequality among contributions of changes to a specific ontology by users, we analyzed the *Normalized Entropy*⁹, which is determined by calculating the *Shannon Entropy* and normalizing the entropy by dividing by the logarithm of the length (i.e., number of users) of a distribution. This coefficient measures the statistical dispersion of a distribution – i.e., the coefficient is one if all users contributed equally to the ontology, while it is zero in case of total inequality where a single user conducts all changes. The results indicate that ICD-11 (0.55) exhibits a low entropy value, i.e., the changes are dominated by only a few users. For NCIt (0.61), OPL (0.64) and ICTM (0.68) we receive medium normalized entropies indicating a more democratic contribution to the ontology by users. A high entropy can be observed for BRO (0.81), which indicates that it is a demographically edited ontology – even though there are only five users.¹⁰

Interpretation & practical implications: The transition probabilities for a first-order Markov chain unveil the roles of certain users and can help to identify users or even groups of users who frequently change the same classes. Users that frequently change classes after other users (i.e., exhibit high transition probabilities in their columns) were identified by us as actual gardeners, curators and administrators of the corresponding projects. If certain users always change the same classes after specific other users, it could be worthwhile for project administrators to investigate if these users are actually collaborating, for example by looking at the changed properties and property

⁹Additionally, we calculated the *Gini Coefficient* for each distribution confirming the results presented here.

¹⁰Note that we do not necessarily know whether the differences between these distributions are statistically significant as we are mainly interested in the behavior of single distributions.

values, or if a single user is always cleaning up after the other user. In all datasets we were able to observe at least one user who contributed a high number of changes, with evenly distributed transition probabilities to all remaining users. This observation indicates that in all projects, gardeners, curators and administrators are assigned (directly or indirectly) certain parts of the ontology; otherwise the transition probabilities between the very active users would be higher.

The ability of understanding who is most likely going to change a specific class next, as well as the classes that a user is most likely to change next could be used by project administrators to help users in finding and identifying classes (and thus work) of interest. On the other hand, the information about the next, most probable contributor for a class, can even be used to create automatic class recommender systems to suggest work to users, which could help to increase participation. However, these two analyses are beyond the scope of this paper and are therefore subject to future work. In particular for projects the size of ICD-11 and NCIt, mechanisms to automatically identify and assign work are highly useful as it is still very time-consuming to find pending work and users with the necessary knowledge to address the identified work-tasks.

4.2. Structural paths

The investigation of *Structural Paths* involves an analysis of different aspects regarding how and where users contribute to the ontology, such as the depth level of the class that users contribute to next (Section 4.2.1) as well as looking at the relationship distances between consecutively changed classes (Section 4.2.2).

4.2.1. Depth-level paths

In this analysis, we investigate if users concentrate their efforts on specific depth levels of the ontology and if there are certain depth levels that are frequently consecutively changed and receive less concentrated workflows. The gathered results provide the necessary information to implement prefetching mechanisms, potentially helping to minimize the loading and waiting times for contributors. Furthermore, we can determine whether users move along the structure of the underlying ontology when editing classes.

Path & model description: For this analysis, we stored the chronologically ordered depth levels of each changed class for each user (user-based). The depth level of a class is the length of the shortest path between the *root node* of the ontology and the corresponding class. For example, a given path for a given user can look like the following: *Depth 3 (for class A), Depth 3 (for class A), Depth 3 (for class A), Depth 3 (for class B), Depth 4 (for class C)*. We merged consecutive changes that were conducted by the same user on the same class into one single sequent change between the same depth levels. Hence, for our previous example we would merge the three successive changes of class A into just two consecutive ones which results in the following final depth-level path: *Depth 3, Depth 3, Depth 3, Depth 4*. This approach helps us to investigate patterns of changing distinct depth levels while still retaining the notion of users consecutively editing the same classes.

Consequently, we fit a first-order Markov chain model on these paths – each path represents a single user and each element of a path represents a corresponding depth level of a class the user has changed. The final transition probabilities give us information about consecutive depth levels that users change over time. For example, they might tell us the probability that users change a class belonging to the third depth level of the ontology after one that has a depth level of 2.

Results: First, the histograms (see top area of Figures 4(a) to 4(e)) show that work is concentrated on certain depth levels of the ontology, with the highest and lowest levels not receiving as much attention as the levels in-between.

As depicted in the transition maps (bottom area of Figures 4(a) to 4(e)), users have a high tendency to edit classes in the same depth levels, visible in the darker colored diagonal. In ICD-11, for the first five depth levels, users appear to have a tendency towards *top-down* editing, evident in the darker immediately right of the diagonal, while this tendency turns around into a *bottom-up* editing behavior, evident in the darker colored squares immediately left of the diagonal, at a depth level of 6 and higher, and appears to be strictly limited to surrounding depth levels. For ICTM (see Figure 4(b)), we can observe a similar trend, again with the tendency towards *top-down* editing appearing to be minimally more dominant. For NCIt, when only looking at the transition map, we can identify a trend towards *bottom-up* editing, evident in the squares directly left of the diagonal being darker than the ones right of the diagonal. However, when also considering the absolute number of changes, depicted in the histogram of Figure 4(c), we can infer that the levels with a higher frequency of occurrence, even though their transition probabilities are more evenly distributed, have a greater impact on the editing strategy. This means that while we can see a *bottom-up* editing behavior for levels 8 to 5 and a *top-down* editing behavior for levels 1 to 4, classes on levels 1 to 4 are more frequently changed than classes on the other levels, hence a tendency towards *top-down* editing can be observed. Thus, when users are not changing the same classes, they still exhibit a preference towards *top-down* editing. Given the short observation periods for BRO and OPL it is hard to infer edit strategies. However, similar to the other projects, we can observe a concentration on the same depth levels with alternating preferences towards higher and lower depth levels. Similar to ICD-11, all datasets exhibit higher transition probabilities between the immediately surrounding depth levels.

Furthermore, we investigate whether the total number of classes as well as the total number of links to the immediate higher (children; edges to classes one level further away from root) and lower (parents; edges to classes one level closer to root) depth level correlate with our findings (Figures 5(f) to 5(j)). For example, the transition map for ICD-11 (see Figure 4(a)) shows that contributors exhibit a *top-down* editing behavior for the first five depth levels, with level 5 exhibiting first signs of *bottom-up* editing. Figure 5(f) shows a higher number of possible transitions from children than parents, indicating that users are in general likelier to follow *top-down* editing-strategies when changing classes, following relationships by chance, of the first four levels. This changes for ICD-11 at level

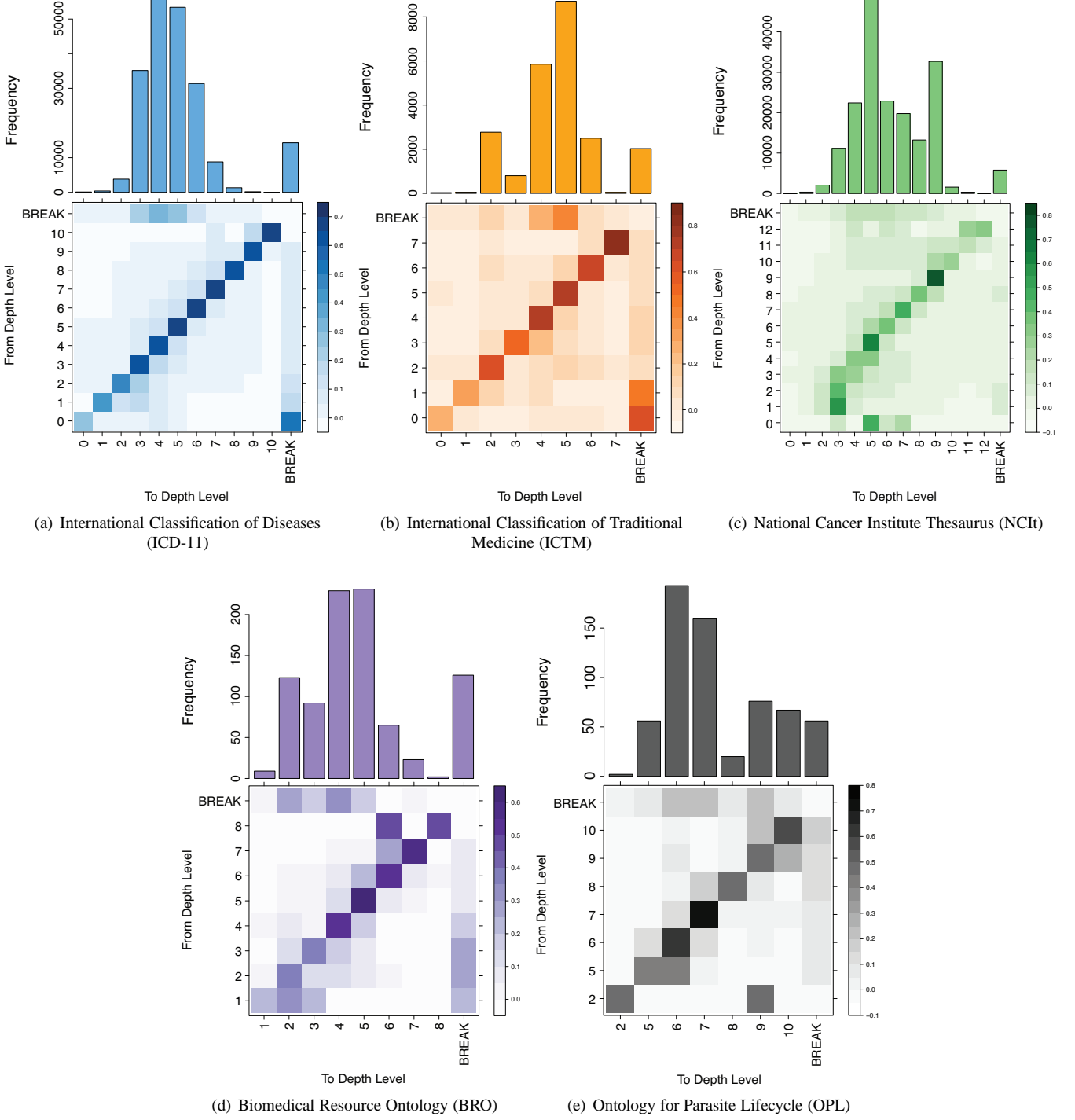
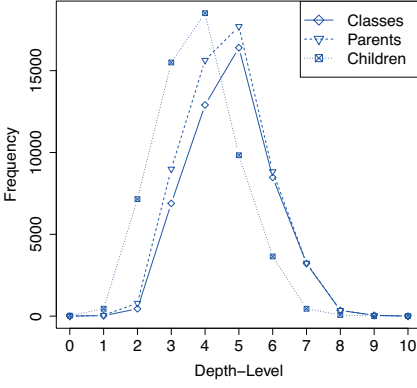
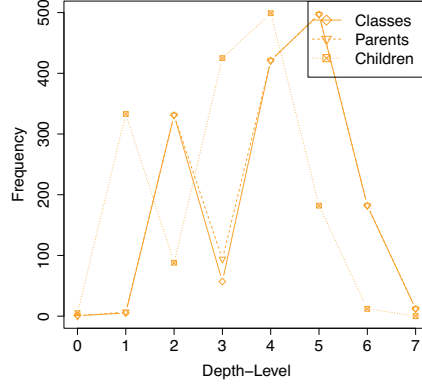


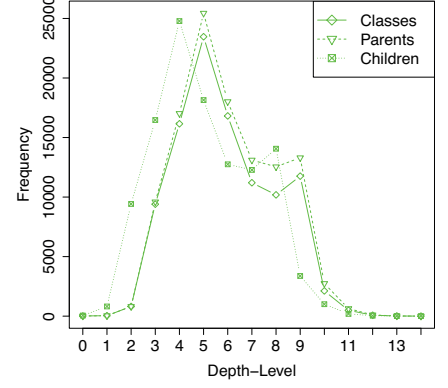
Figure 4: Results for the Depth-Level Paths analysis: The columns and rows of the transition maps (**bottom area** of Figures 4(a) to 4(e)) represent the transition probabilities of a first-order Markov chain between depth levels, where rows are *source depth levels* and columns are *target depth levels*. A sequence (or transition probability) is always read *from row to column*. Darker colors represent higher transition probabilities while lighter colors indicate lesser transition-probabilities. Absolute probability values are dependent on the number of investigated rows and columns, hence relative differences are of greater importance. For classes closer to root a *top-down* editing manner can be observed, while this is reversed for classes further away from root. The sum of all transition probabilities for one row in the transition maps is 1. For example, if *Depth-Level 6* exhibits a transition probability of 0.30 to another *Depth-Level 5* it means that a class on *Depth-Level 5* has a 30% probability of being changed after a class on *Depth-Level 6*. The histograms (**top area** of Figures 4(a) to 4(e)) show the number of changes performed in each depth level aggregated over all users of the respective projects (again for a first-order Markov chain). Throughout all projects, classes located between the first and last few depth levels (in the middle) are changed substantially more frequently than others, suggesting that work is concentrated on some depth levels while others receive none to very few changes at all. Note, that the y-axes for all histograms are scaled differently for each dataset. For the x-axes (and column/rows of the transition maps) we only display depth levels which exhibit at least one change, thus, the depth level sequences are not necessarily continuous from lowest to highest depth level.



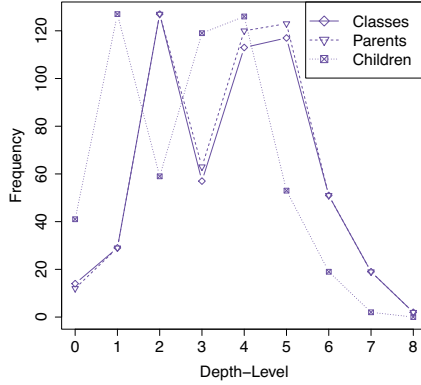
(f) International Classification of Diseases (ICD-11)



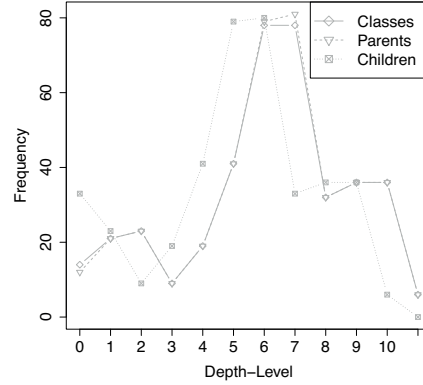
(g) International Classification of Traditional Medicine (ICTM)



(h) National Cancer Institute Thesaurus (NCIt)



(i) Biomedical Resource Ontology (BRO)



(j) Ontology for Parasite Lifecycle (OPL)

Figure 5: The **Figures 5(f) to 5(j)** depict the absolute numbers (y-axis; Frequency) of classes as well as the number of edges (*isKindOf*) to classes on the immediate higher (*parents*; closer to root) and lower (*children*; further away from root) depth level for all depth levels (x-axis; Depth-Level). According to Figures 5(f) to 5(j) the transition probabilities depicted in the transition maps correlate with the total number of edges to children and parents for each depth level across all datasets.

5, with a higher number of transitions to parents than to children, and continues until level 10. Resulting in a higher probability of users performing *bottom-up* editing-strategies when changing classes from levels 6 to 10. The same observations can be made for all other datasets, indicating that the class hierarchy influences the edit behavior of contributors.

In all datasets, after taking a *BREAK* (representing an artificially introduced session break when two consecutive changes of the same user are more than 5 minutes apart; for more information see Section 5.4), users exhibit a clear tendency towards changing classes on certain depth levels (e.g., levels 3 to 5 for ICD-11, levels 4 to 5 for ICTM, levels 4 to 7 for NCIt, levels 2 to 4 for BRO and levels 6 to 9 for OPL).

Interpretation & practical implications: The results of this analysis show if, to what extent and where (limited to locality being determined by *isKindOf* relationships) work is conducted and concentrated within the ontology. This information can potentially be used in a variety of ways, for example by ontology-engineering tool developers to adapt the interface of the ontology-engineering tool dynamically to display specific

classes after users return from a *BREAK*. Project managers can adapt milestones and project progress reports to reflect the underlying editing strategies (e.g., *top-down* editing), for example by aligning progress with created branches (opposed to complete coverage). Another potential use-case for the results of this analysis involves the prefetching of content in certain environments (e.g., mobile or embedded systems) to minimize waiting times. Across all projects we can observe that classes close to and very far away from the *root* of the ontology are not edited as frequently as other classes. One explanation for this observation could be that classes in lower depth levels (closer to *root*) are mainly used as content dividers and are usually created in the beginning of a project. Thus, they may be more stable and less frequently updated. Classes at the higher depth levels (further away from *root*) on the other hand most likely require extensive expert knowledge. Hence, only a small number of users have the necessary expertise to contribute to these classes. Additionally, the absolute number of classes in the higher and lower depth levels is much lower in all investigated datasets. Note that absolute values of depth levels are less important for

the interpretation of the results than their relative position (i.e., closest to root, furthest away from root, etc.). For example, a class at level 6 can exhibit different behaviors in ontologies with 6 or 10 levels.

In all projects, except for NCI, the depth levels where users start to edit the ontology after they return from a *BREAK* are similar to the ones where they stop editing before taking a *BREAK*. To be able to make that observation we have to take the absolute numbers of changes on each depth level (bottom area of Figure 4) into account when looking at the transition probabilities (top area of Figure 4). NCI is the only dataset where users appear to be similarly likely to take a *BREAK* after changing classes across all depth levels, except for 0 and 12.

When we combine the results of this analysis with the results of the *User-Sequence Paths* (Section 4.1) we may be able to develop automatic mechanisms to curate and delegate work to users. For example, if we know that a specific user is most probably going to contribute to a class on level 3 and we have a set of classes on that level where that specific user is the most probable next user to contribute to, determined by the *User-Sequence Paths* analysis, we may combine these two observations to create class (and thus work) suggestions for users.

4.2.2. Hierarchical relationship paths

Given the high number of observed transitions between the same depth levels in the *Depth-Level Paths* analyses (Section 4.2.1; bottom area of Figure 4), we conducted an additional analysis investigating the relationships between the changed classes for all users. Hence, we wanted to know if all worked-on classes on the same depth-levels are siblings, cousins or any other kind of close relative? And in general, can we determine if users follow these hierarchical orders of an ontology when contributing to classes on the same depth level? To further strengthen our observation that users are actually moving along the ontological hierarchy when contributing to an ontology (see Section 4.2.1), we analyzed the relationships between the changed classes for each user. Note that whenever we talk about relationships for this analysis, we refer to the hierarchical *isKindOf* relationships between two classes, e.g., parent, child, sibling or cousin. For example, when traversing the shortest-path distance of 2, multiple different nodes can be reached, such as a grandparent (i.e., 2 times up), a grandchild (i.e., 2 times down), a sibling (i.e., 1 time up, 1 time down) or even some other relationship (e.g., 1 time down, 1 time up).

Path & model description: By combining the information from the *Depth-Level Paths* and the relative movement between depth levels, we inferred the hierarchical relationships between two consecutively changed classes of a single user (user-based). For example, if the difference between the depth levels of the investigated classes would be exactly the size of the shortest-path between them (with the shortest-path being > 0), the latter-changed class could either be a *Child*, a *Parent*, an *Ancestor* or a *Descendent* of the first-changed class. Given a relative *DOWN* movement (to a lower depth level) value, depending on the shortest-path value, the second class could be classified as *Child* (shortest-path of 1) or *Descendent* (shortest-path > 1). Analogously follows the definition of a *Parent* and *Ancestor* with a

relative *UP* movement. A *Sibling* is defined as the two classes being (i) connected via the same parent with (ii) a shortest-path distance of 2 and (iii) both classes are located on the *SAME* depth level. A *Cousin* is used when two classes on the *SAME* depth level are connected by the same grand parent while exhibiting a shortest-path distance of 4. Every other possible combination of depth level and shortest-path was classified as *Other*. *Self* indicates that the same class that was changed last time was changed again. For example, a consecutive change of *Sibling* and *Self* means that a change was first performed on a class that is a sibling of the previous class (not displayed in this example) and then another change was performed on the same class, however now the relationship changed to *Self* as no new class was involved.

Again, consecutive changes on the same class by the same user have been merged into one single sequent change (c.f. Section 4.2.1), meaning that multiple (more than 2) consecutive changes of the same user on the same class have been merged into *Self* to *Self*. Hence, a given path for a single user can, e.g., look like the following: Sibling, Self, Self, Child.

We fit a first-order Markov chain model to the data – each path represents a single user and each element represents a hierarchical relationship between the classes changed by the user. The resulting transition probabilities of the fitted model can then give us insights into common emerging patterns. E.g., we can identify how probable it is that users change a *Sibling* after a *Child*.

Results: When looking at the histograms (see top area of Figures 6(a) to 6(e)), we can observe that the relationships *Self*, *Sibling* and *Other* are highly represented across all datasets. The transition maps (bottom area of Figures 6(a) to 6(e)) show that after a *BREAK*, across all five datasets, users tend to change classes “somewhere else” in the ontology, evident in the high transition probability from *BREAK* towards *Other*, and are likely not to resume work in the same area of the ontology that they stopped working on. For ICD-11, ICTM and OPL, no matter which relationship type occurs, users tend to edit the same class consecutively (dark colors in the *Self* column). From this *Self* relationship, which is also the one that occurs the most often in ICD-11, ICTM and OPL, users are very likely either to change the same class again (*Self*) or to change a *Sibling* of the current class.

For NCI, BRO and OPL we can observe that users, when changing a *Parent* are very likely to change a *Child* of that parent afterwards. Note, that this *Child* does not necessarily have to be the same class that was changed prior to the traversal to *Parent*. In all datasets, except for OPL, very high transition probabilities towards *Other* can be observed for all not so frequently present relationships. In particular for NCI we can observe that *Other* is the most frequently observed transition, even before *Self* and *Sibling*.

Interpretation & practical implications: By combining the results of this analysis with the results of the *Depth-Level Paths* analysis, we can infer that users exhibit a tendency towards *top-down* editing while contributing to the ontology, when only considering changes that occur on different depth levels. If they concentrate their efforts on the same depth levels, users

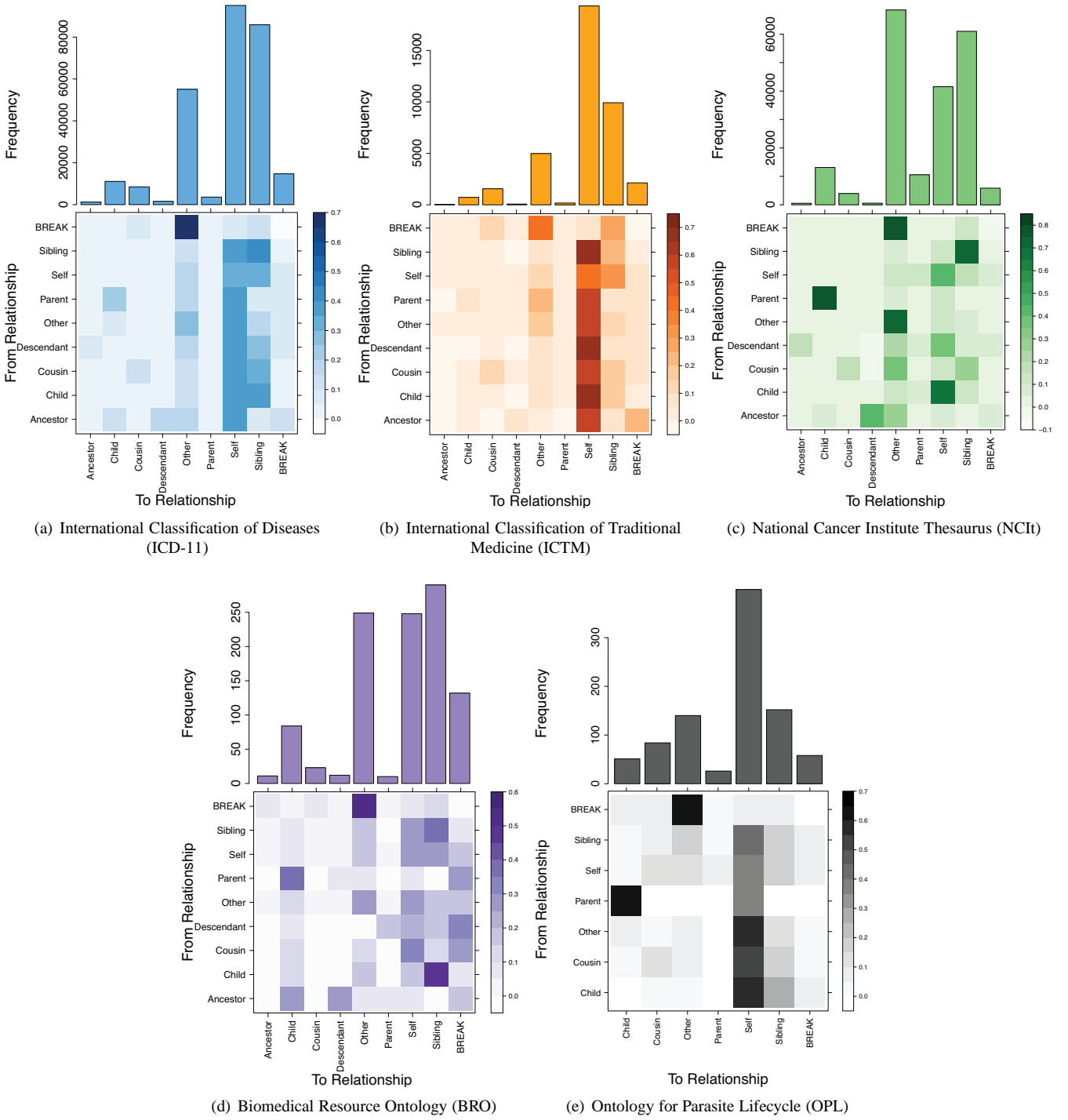


Figure 6: Results for the Hierarchical-Relationship Paths analysis: The columns and rows of the transition maps (**bottom area** of Figures 6(a) to 6(e)) represent the transition-probabilities of a first-order Markov chain between hierarchical-relationship levels, where rows are *source relationships* and columns are *target relationships*. A sequence (or transition-probability) is always read *from row to column*. Darker colors represent higher transition-probabilities while lighter colors indicate lesser transition-probabilities. Absolute probability values are dependent on the number of investigated rows and columns, hence relative differences are of greater importance. Across all datasets, aside from *Self*, a very clear trend towards editing the ontology along *Siblings* can be observed. The histograms (**top area** of Figures 6(a) to 6(e)) show the total number of occurrences of each relationship in the corresponding datasets aggregated over all users (again for a first-order Markov chain). Note, that the y-axes for all histograms are scaled differently for each dataset. For the x-axes (and column/rows of the transition maps) we only relationships that occur at least once in the corresponding paths, thus the x-axes could be different from project to project. Given the very high amount of *Self* and *Sibling* transitions we can concur that users, when they contribute to classes on the same depth level follow a *breadth-first* strategy, meaning that they first concentrate their work on closely related classes (*Siblings*) on the same depth-level before switching to a different branch on the same or any other depth-level.

exhibit a *breadth-first* editing behavior, meaning that they first concentrate their work on closely related classes (*Siblings*) on the same depth-level before switching to a different branch on the same or any other depth-level, either changing the same class multiple times or traversing along siblings of the current class. We can leverage this information not only to refine the previously suggested pre-fetching of classes but also to enhance possible class recommendations. Similarly, it is possible for ontology-engineering tool developers to minimize the necessary efforts of users to contribute to the ontology by implementing, for example, guided workflows that take the underlying edit strategies of the contributors into account.

As classes in ICD-11 and ICTM have a large number of properties and for ICTM certain properties have to be added in multiple languages, the high transition probabilities towards *Self* (dark colors in the *Self* column) are not surprising. One possible explanation for this observation for ICD-11 could be the special functionality available in iCAT (for ICD-11) that allows users to export parts of the ontology as spreadsheets for local editing and adding property values. Once contributors finished editing the spreadsheet they have to enter the data into the system manually, as no automatic import functionality is present. In the iCAT interface, users are simultaneously presented with the ontology tree for navigating through the classes and the corresponding properties and property values. When users select a property they can easily switch between classes, with the selected property staying selected, thus allowing to quickly enter the same properties for different classes.

A similar, yet not as dominant as in ICD-11 and ICTM, behavior can be observed for NCIt and BRO and even to some extent in OPL, which all do not use the export functionality. According to our observations, users travel along the underlying hierarchy when contributing to the ontology. Given the observations made for ICD-11 this behavior can be enforced by providing certain functionalities in the user-interface especially when they compliment the workflows of the contributors.

The results of this analysis have also shown that users are likely to pursue a certain strategy or intermediate goal for their edit sessions, for example changing all classes in a specific (narrow) area of the ontology. This is evident in the observation that after returning from a *BREAK*, users have a very high tendency to change the ontology “somewhere else” (see the transition probabilities from *BREAK* towards *Other* in the top-row of Figure 6), rather than picking up the work, where they left off. This discovery is very important for developing class-recommender, as we may use the results of this analysis to suggest closely related classes to the current class a user is working on, however when that user stays inactive for the duration defined for introducing *BREAK*s the recommendation strategy has to be changed.

4.3. Property paths

Aside from analyzing different aspects of activity (Section 4.1) and the correlation between contribution patterns and the structure of an ontology (Section 4.2), we can use Markov chains to perform an analysis on the properties that are consecutively changed by users in an ontology. This means that, for example,

if a property value was edited by a user, we extracted the property (not the value) and created chronologically ordered lists of properties, whose values were changed by the corresponding users. For example, if a user changed the title of a specific class, we would extract *title*, rather than the value inserted into the title property. Now, we provide insights into emerging patterns from different viewing angles for the observations. Thus, we look at property sequences for (a) single users (user-based) and for (b) single classes (class-based) – see Section 3.2. We were not able to perform the *Property Paths* analysis on OPL and BRO as these datasets contain only a very limited number of unique property value changes during our observation periods. We also had to discard the results from NCIt, as the ontology-editing environment for NCIt provides a unique change-queuing mechanism that allows for multiple property values to be changed at the same time, making it impossible to extract chronologically ordered sequential property patterns.

Path & model description: First, we extracted the properties whose values were changed in ICD-11 and ICTM, sorted either by user and timestamp or by class and timestamp. Finally, two different types of chronologically ordered property lists were extracted, one ordered per user and one ordered per class (for both datasets). The properties in *Property Paths* represent the ones which can be assigned a value for each class in ICD-11 and ICTM. Whenever a change did not modify a property (e.g., because the change action dealt with moving or creating a class) we added the element *no property* to the corresponding path. A potential path for a single user or class then may look like: *title, title, title, use*. Similar to previous analyses, if the same user has consecutively changed the same property (e.g., in the previous example *title*) on the same class, we merged these multiple changes into one successive change. Analogously, however without the restriction of the same user, if the same property was changed on the same class, we merged these changes into one sequent change. For previous example, if changes would have been performed editing the referenced properties for a single class, we would end up with the path: *title, title, use*.

Consequently, we fit a first-order Markov chain model on this set of paths (for users or classes). The final transition probabilities of the model then give us information about the probability of changing a value of one property Y after another property X either for users or for classes. For instance, we can find the property Y that most frequently has been changed after property X for classes.

Results: When looking at the histograms (top area in Figures 7(a) to 7(d)) we can see that even after removing not very frequently used properties,¹¹ both datasets exhibit a few properties which have received a high number of changes, while the remaining majority of properties only received a very limited number of changes. For both datasets, aside from *no property*, the properties *use*, *title* and *definition* appear to be the most frequently used properties. As can be seen in the top area

¹¹ All properties which were rarely edited have been removed from Figure 7 as they do not hold information but their removal increased the readability of the plots dramatically.

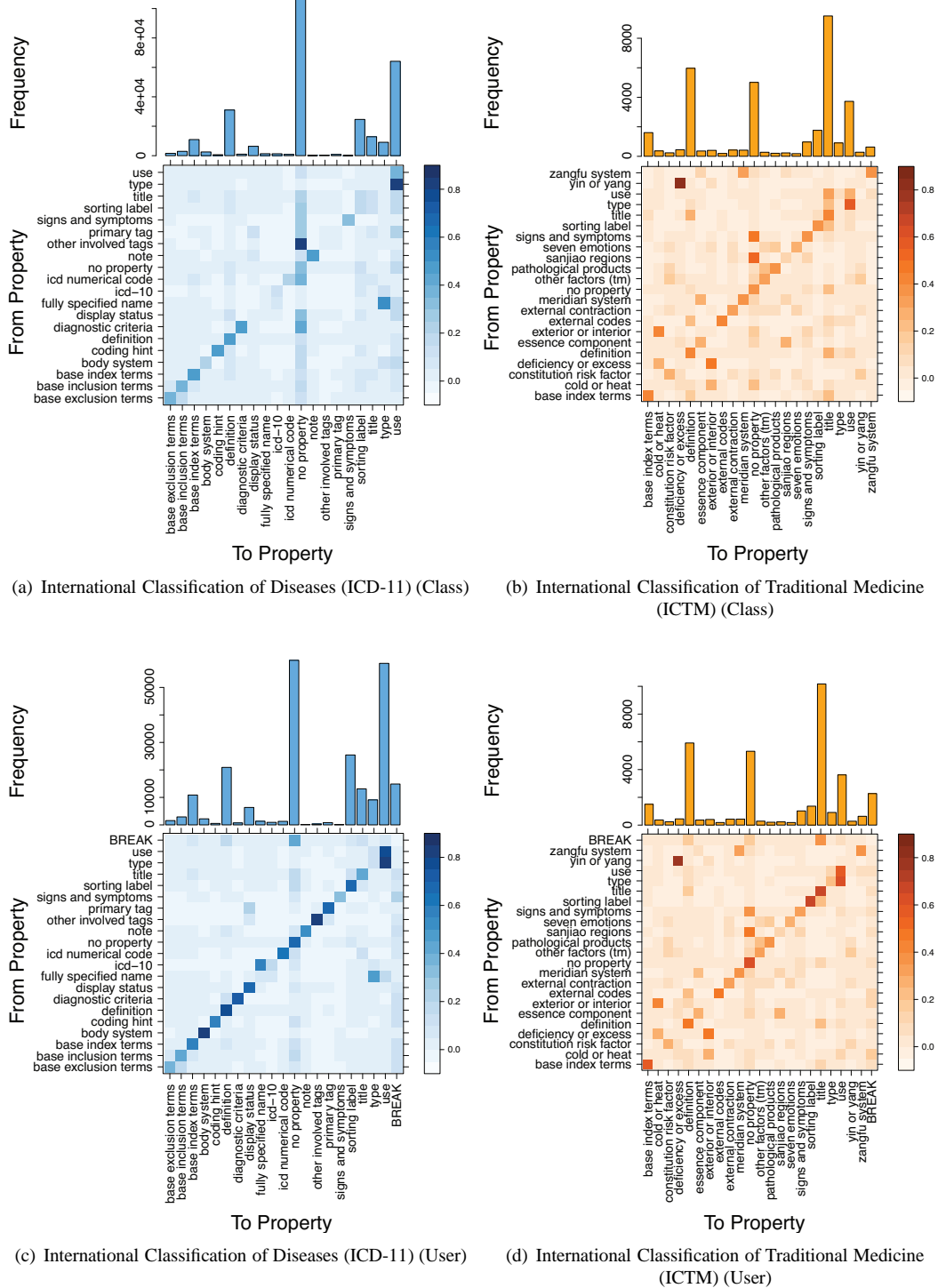


Figure 7: Results for the Property Paths analysis: The columns and rows of the transition maps (**bottom area** of Figures 7(a) to 7(d)) represent the transition-probabilities of a first-order Markov chain between consecutively changed properties, where rows are *source properties* and columns are *target properties*. Figures 7(a) and 7(c) represent class-based patterns while Figures 7(b) and 7(d) visualize user-based patterns. A sequence (or transition-probability) is always read *from row to column*. Darker colors represent higher transition-probabilities while lighter colors indicate lesser transition-probabilities. Absolute probability values are dependent on the number of investigated rows and columns, hence relative differences are of greater importance. Across all datasets a very clear trend towards consecutively editing the same properties can be observed. The histograms (**top area** of Figures 7(a) to 7(d)) show the total edits of each property in the corresponding datasets aggregated over all users and classes (again for a first-order Markov chain). Note, that the y-axes for all histograms are scaled differently for each dataset. As ICTM and ICD-11 only share a limited amount of properties the x-axes (and column/rows of the transition maps) are different from project to project. In both projects and across all 4 different approaches the *title*, *definition* and *use* properties are frequently used. Due to reasons of readability we were forced to remove properties from the plots, which exhibited only a very limited number of changes, thus did not provide substantial information for the purpose of this analysis.

of Figures 7(a) and 7(b), multiple consecutive changes of the same property appear to be fairly common for both datasets. In contrast, when looking at Figures 7(c) and 7(d), which depict the transition probabilities between the sequences of properties changed by each user, we can see an even stronger trend towards consecutively changing the same properties across different classes, especially *definition*, *title* and *use*. For ICD-11 Figures 7(a) and 7(c) show that the class-based approach is less focused on consecutively changing the same property, evident in the brighter diagonal, when compared to the user-based approach. This is due to the export functionality available in iCAT combined with the manual process of inserting the same property for different classes by users of ICD-11. In contrast, such functionality is absent in ICTM, thus leading to similar behaviors for the class and user-based approaches for ICTM. The fact that a large portion of successive changes are conducted on the same property for both approaches analyzed for ICTM could also be due to the multilingual nature of the project, meaning that certain properties, such as *title* and *definition*, have to be entered multiple times in multiple languages. Similar results have been presented by Wang et al. [24], who used association rule mining techniques to analyze the change-logs of ICD-11 and ICTM.

Contributors in ICD-11 have a high tendency of performing *no property* changes after they return from a *BREAK* followed by *use*, *title* and *definition*. In ICTM, users resume their work primarily by changing the *title* property, the *definition* property followed by *no property* changes.

Interpretation & practical implications: One of the main benefits of this analysis is the identification of commonly and consecutively changed properties for classes and users. In turn, this information might potentially be used to suggest work (e.g., prompting a user to check a certain property by combining the *User-Sequence Paths* analysis and the *Property Paths* analysis), or by ontology-engineering tool developers to potentially anticipate the property a user is most likely to change next. The fact that classes appear to exhibit more diverse property-contribution patterns when being changed than users could be a direct result of the multi-lingual nature of ICTM and the already mentioned export functionality present in iCAT. This means that given the most recent property of a class that was edited, we may predict which property is most likely to be changed next. Similarly, we can predict the property a user is going to edit next.

5. Findings and discussion

In this section we first summarize our findings in Section 5.1 before we shortly discuss the potential applicability of higher order Markov chain models in Section 5.2. Next, we discuss differences between the investigated projects in Section 5.3 and finally, point out potential limitations of this work in Section 5.4.

5.1. Summary of findings

We will now discuss our main findings (Table 2) and explore their consequences.

Emergence of micro-workflows: By investigating whether sequential user-contribution patterns (see Section 4.1) can be identified in five different collaborative ontology-engineering projects, we have shown that users appear to work in micro-workflows, indicating that for all investigated projects, each user contains predictive information about the user, who is going to contribute to a specific class next.

Additionally, however not presented in this paper due to reasons of space, we have also conducted an analysis to determine the change type (e.g., adding a property value, moving a class, replacing a property value, etc.) a user is most likely to perform next (as shown in Walk et al. [30] for ICD-11). In this analysis we were able to extract a first-order Markov chain for all datasets presented in this paper, meaning that the last change type that a user performed contains information about the next change type of that user. When combining the information about the user who is most likely to contribute to a class next and the specific change action that this user is most likely to conduct (or the change action that is most likely conducted on a class next), we can create specific tasks for contributors, asking them to perform a certain change on a specific class.

Our results could be used by project managers and ontology-engineering tool developers to identify classes for users and users for classes, helping editors to minimize the necessary efforts for finding and identifying classes to contribute to. Moreover, automatic means of curating and delegating work-tasks to users can be derived by ontology-engineering tool developers, which can help to potentially increase participation as discussed in Kittur and Kraut [31].

User roles can be identified: Across all datasets we were able to identify that a limited number of users have contributed to the majority of all changes. These highly active users are very likely to be *target users* for all other users, meaning that they are very likely to change the same class after another user. Across all five datasets, the roles of these *target users* could be identified by us as moderators or administrators of the corresponding projects performing maintenance tasks, such as gardening (e.g., pruning outdated classes, fixing errors, etc.) or manual verification of newly added data.

Furthermore, we were able to show that moderators and administrators divide work among each other, as they are not very likely to change the same classes directly after another administrator or moderator, even though these users exhibit the highest absolute numbers of changes in the corresponding projects. Looking at the transition probabilities of Figure 3 it is possible to identify users or even groups of users who have a high tendency to work on the same classes, thus might be collaborators or reverting/correcting changes of each other.

Users edit the ontology top-down and breadth-first: The *Depth-Level Paths* analysis (see Section 4.2.1) demonstrated that users have a very high tendency of staying in the same depth level when contributing to the ontology. If editors change depth levels while editing the ontology they exhibit a minimal preference to do so in a *top-down* rather than a *bottom-up* manner. Furthermore, the results suggest that users move along the hierarchy as we were able to show that they follow a *top-down* editing strategy for classes that are closer to the root node while

Table 2: A summary of all findings applicable to all investigated biomedical ontologies. All listed findings are discussed in more detail in Section 5.

User-sequence paths (cf. Section 4.1)	Users work in micro-workflows	Information about which users successively change a class can be identified; i.e., information about who has edited classes in the past contains predictive information about who is going to change a class next.
	User-roles can be identified	Looking at historic data, we can identify different user roles, i.e., administrators and moderators, gardeners (a contributor focused on pruning ontology classes and fixing syntactical errors) and users that frequently interact with (collaborate/revert) each other.
Structural paths (cf. Section 4.2)	Users’ edit behavior is influenced by the class hierarchy	Contributors, when adding content to the ontology, are influenced by the class hierarchy.
	Users edit the ontology top-down and breadth-first	By and large, users exhibit a minor tendency towards top-down editing behavior when changing hierarchy levels while contributing. However, when staying in the same hierarchy level, contributors rather follow a <i>breadth-first</i> edit behavior, moving from one sibling of a class to the next sibling.
	Users edit closely related classes	Contributors have a very high tendency to consecutively change closely related classes, as opposed to randomly and distantly related classes.
Property paths (cf. Section 4.3)	Users perform property-based workflows	Contributors, when adding content to the ontology, tend to concentrate their efforts on one single property, which is added and edited for multiple classes.

this changes to a *bottom-up* editing strategy for classes closer to the deepest depth levels and transitions are more likely to occur along the immediate higher or lower depth level.

To further investigate the distances between changed classes at the same depth levels we investigated the *Hierarchical Relationship Paths* (e.g., child, parent, sibling, cousin, etc.) between these changed classes. We found that users, when they edit classes on the same depth level, follow a *breadth-first* manner, focusing on editing all the siblings of a class before switching to a completely different area of the ontology to continue their work after a *BREAK*.

Users edit closely related classes: Additionally to the *breadth-first* manner that users follow when editing classes in the same depth level, we discovered that users have a very high tendency to work on closely related classes (e.g., the sibling or cousin of the currently changed class). The information collected in Section 4.2 allows to potentially predict (or narrow down) the class a user is going to contribute to next, which, if accurate, is a very valuable information that could be used for a variety of improvements and adaptations. For example, project-administrators could adjust the milestones of the development-strategy to better reflect the way users contribute to the ontology while user-interface designers could emphasize certain areas of the ontology to direct users towards specific classes – especially after they return from a *BREAK* – or implement pre-fetching algorithms to minimize load-times. For contributors in particular, the task of identifying and finding classes that they (i) want and (ii) have the necessary expert knowledge to contribute to is a time-consuming task, which potentially can be minimized by implementing class recommender based on the results of the *Structural Paths Analysis* and *User-Sequence Paths Analysis*.

Users perform property-based workflows: The investigation of sequential patterns for property-contributions showed that in ICD-11, users have a very high tendency of consecutively changing the same property across multiple classes. We could also identify specific patterns that emerge when users suc-

cessively change properties in collaborative ontology-engineering projects.

The results collected in the Section 4.3 provide new insights for administrators and ontology-engineering tool developers, as they allow the generation of work-tasks (e.g., Please verify the property *title* of the class *XII Diseases of the skin!*). So far, users are always presented first with the section of the interface that allows for changing or adding the *title* and *definition*, which could be one explanation for the high probabilities of users changing these properties when returning from a *BREAK*.

Note, that for this analysis we have used the data from ICD-11 and ICTM, which both share a very similar ontology-engineering tool, thus the results might be biased towards the used ontology-editor.

5.2. Higher order Markov chains

Based on our proposed methodology of using first-order Markov chain models (see Section 3.3) resulting in the findings summarized in Section 5.1, we currently lay our focus on detecting patterns only derived from successive interactions within collaborative ontology-engineering projects. This means, that we identify how likely it is that one specific interaction follows another one (e.g., which user edits a class after another one). This is reasoned by the definition of a first-order Markov chain based on the Markovian property which postulates that the next interaction only depends on the current one.

Contrary, Markov chain models can also be defined on higher orders; this means that the next state of the model (or interaction in our case) depends on a series of preceding ones instead of only the current one. For example, a *second-order* Markov chain model postulates that the next state depends on the current state and also the previous one. Previous studies suggest that human navigation on the Web might be better modeled by using higher order models compared to first-order models (e.g., [32, 29]). Hence, we could assume that this might also be the case for our use-case. By also modeling our data with such

higher order models, we would potentially be able to identify longer patterns (e.g., *User A* regularly edits a class after *User B* and *User C*). Also, possible recommender systems could benefit from the additional predictive power of such higher order chains.¹² While highly interesting, this analyses would be out-of-scope for this article which is why we leave this open for future work.

5.3. Differences between the investigated projects

Even though each project exhibits a different number of depth levels, which all receive a different amount of attention by the contributors, we can observe commonalities of edit strategies between them. For example, the levels 3 to 6 exhibit the highest number of changes in our observation period for ICD-11, while for OPL these levels are 6 and 7.

Regarding the hierarchical relationships we can see that consecutively changing the same class is very likely to happen in ICD-11, ICTM, BRO and OPL regardless of the source relationship (evident in the darker colored *Self* columns in Figures 6(a), 6(b), 6(d) and 6(e)). This *Self*-relationship is still very prominent, however the transition probabilities towards *Self* for NCIt are not as dominant as they are for the other datasets.

Another observation depicted in the transition maps is the clear focus on transitions from *Sibling* to *Sibling* across three out of five datasets, with the exception of ICTM and OPL. One explanation for ICTM could be the fact that some properties of the ontology are multi-lingual, thus require users to add multiple languages for the same property, which are all stored as a single change. For OPL, transitions, except towards *Self* are in general really scarce, indicating that users focused on editing and entering multiple property values (or one property value) of a single class before continuing to the next class.

When looking at the sequence of changed properties for each class (in contrast to: for each user) we can observe a concentration on consecutively changing the same property in ICTM, which is most likely a direct result of the multi-lingual nature of the properties used in this project. In ICD-11 on the other hand, transitions between changed properties of classes are much more diverse and less focused on transitions between the same properties. This observation indicates that either not all properties have received a substantial amount of values for all the possible properties and/or that users make use of this special export functionality of iCAT, thus successively changing the same property is less common as the content is only inserted once into the system.

In the *User-Interface Sections Paths* analysis we have mapped the changed properties to the corresponding sections of the user interface of the used ontology-engineering tools, which essentially represents a more abstract analysis of the *Property Paths* analysis. By investigating the sequences of user interface sections we could confirm that, for ICD-11, users have a very high tendency to consecutively change the same properties for multiple classes, evident in the scarce transitions between different

sections and the high concentration on transitions between the same sections. For ICTM this behavior was not as distinctive as it was for ICD-11, which could be due to the missing export functionality and therefore the lack of the previously explained manual import sessions.

In general these observations indicate that the absence or presence of a given functionality of the ontology-engineering tool can produce (and influence) different editing behaviors when developing an ontology.

5.4. Limitations

We were not able to recreate the exact class hierarchy of the ontology for every single change across our observation periods for all datasets. This limitation is partly due to a lack of detail in the change-logs. Thus, we decided to focus our analysis, using all five ontologies *as is* at the latest point in time, which is also what would most likely be used in a *real-world* scenario.

For example, if a class was changed by a user while it was located on depth level 3 and at a later point in time moved to a different location where it now resides at depth level 5, we would assume that this class has always been on depth level 5. Please note that this bias is only present in the *Structural Paths* analyses (Section 4.2). To measure the extent of the potential bias, we counted all changes that were performed on a class before it was moved within in the ontology. Applying this rule to our change dataset, we collected a total of 116, 204 of 439, 229 changes for ICD-11 and 18, 958 of 67, 522 for ICTM. These numbers represent about 1/4 and 1/3 of all changes for ICD-11 and ICTM respectively. For BRO 276 of 2, 507 (ca. 1/10) and for OPL 2 of 1, 993 of all changes were performed on classes, which have been moved afterwards.

Note that an additional requirement for the identification of sequential patterns in collaborative ontology-engineering projects using Markov chains is the availability of rather large change-logs. In general, the less common entities (e.g., properties) are present in the change-log the more (exponentially) observations have to be available in order to detect more fine-grained patterns. Without enough observations (changes), the identification of sequential patterns is either very hard, and can only be approximated, or not possible at all. As can be seen in Table 1, we have selected all of our datasets to satisfy this requirement, as all chosen datasets exhibit a substantial number of changes.

Furthermore, we have included *artificial session breaks* into our analysis as described by Walk et al. [30] to analyze where or what users start to edit in the ontology and where or what users edit before they take a break. For all user-based analyses we have introduced a *BREAK* if two consecutive changes of the same user were apart longer than 5 minutes.

All analyses in this paper are based on *isKindOf* relationships for determining distances and locations within the ontology. We plan on further expanding this analysis by investigating the impact of other kinds of relationships and other features that are available in ontologies on our pattern detection approach.

Even though all datasets presented in this paper are created with WebProtégé or one of its derivatives, there is only one requirement that prevents practitioners from performing this analysis on other ontologies: The availability of a change-log (in

¹²Note that it is necessary to apply model selection techniques as described in [29] in order to identify the most appropriate Markov chain order based on statistical significant improvements of higher orders compared to lower orders

the required granularity for the deemed analyses) that can be mapped onto the underlying ontology. Note that it would be possible to conduct this analysis for ontologies created by single individuals, meaning that “collaboration” is only a requirement when the nature of the analysis requires investigating transitions between multiple users.

Also, the kind of knowledge base (classification, taxonomy or ontology), the used representation language (e.g., OWL and OWL-DL expressivity, RDF, Turtle) or the development tool of a particular collaborative ontology-engineering project in question does not prohibit conducting a pattern analysis as presented in this paper, as long as the underlying knowledge base (and thus the change-log) exhibits the necessary granularity and the semantic properties of interest for the analysis.

However, this also means that the differences of the knowledge representation used languages (i.e., expressivity and types) are not considered by our analysis, with NCI being a thesaurus and the rest of the investigated datasets being ontologies. Thus, whenever differences are observed between NCI and the remaining datasets, further research is warranted to determine the origin of this observation.

Furthermore, the analysis presented relies on investigating usage logs of collaborative ontology-engineering projects by looking at changes, performed by users of the corresponding systems. As this only represents one possible way of interacting with the underlying ontology, albeit the most frequently used one, an extension of the conducted Markov chain investigation warrants future work to include, for example, discussions for consensus building, suggestions of terms by users or automatic imports.

6. Related work

For the analysis and evaluation conducted in this paper, we identified relevant information and publications in the domains of (i) Markov chain models, (ii) collaborative authoring systems and (iii) sequential pattern mining.

6.1. Markov chain models

In the past, Markov chain models have been heavily applied for modeling Web navigation – some sample applications of Markov chains can be found in [33, 34, 35, 36, 37, 38]. Also, the Random Surfer model in Google’s PageRank [39] can be seen as a special case of a Markov chain.

Previously, researchers investigated whether human navigation is memoryless (i.e., of first order) in a series of studies (e.g., [40, 36]). However, these studies mostly showed that the memoryless model seems to be a quite plausible abstraction (see e.g., [41, 42, 37, 38]). Recently, a study picked up on these investigations and suggested that the Markovian assumption (i.e., property) might be wrong [32]. However, this study did not reveal any statistically significant improvements of higher order models. Singer et al. [29] solved this problem by developing a framework for determining the appropriate order of a Markov chain for a given set of input data. In Walk et al. [30] we applied and mapped the presented framework onto structured logs

of changes and provided an in-depth description of the requirements and steps necessary to use the framework in this setting.

In this paper we present a detailed analysis of sequential patterns by applying and analyzing Markov chains across the change-logs of five collaborative ontology-engineering projects in the biomedical domain. A more detailed explanation of the necessary steps to be able to apply Markov chains onto the change-logs of collaborative ontology-engineering projects is presented in Walk et al. [30]. Note that we focus on applying first-order Markov chain models in this work while we see the application of also higher order models as highly interesting future work as discussed in Section 5.2.

6.2. Collaborative authoring systems

Research on collaborative authoring systems such as Wikipedia has in part focused on developing methods and studying factors that improve article quality or increase user participation. These problems represent important facets of collaborative authoring systems and solutions to tackle these problems are of interest for collaborative ontology-engineering projects.

For example, Cabrera and Cabrera [43] demonstrated the effect of minimizing the costs and efforts necessary for users to contribute on potentially achieving higher contribution rates. Another approach, also presented by Cabrera and Cabrera [43], focuses on providing an environment where interactions and communication between contributors are encouraged and performed frequently over a long period of time to establish a group identity and to promote personal responsibility.

More recent research on collaborative authoring systems, such as Wikipedia, focuses on describing and defining not only the act of collaboration amongst strangers and uncertain situations that contribute to a digital good [44] but also on antagonism and sabotage of said systems [45]. It has also been discovered only recently that Wikipedia editors are slowly but steadily declining [46]. Therefore Halfaker et al. [47] have analyzed what impact reverts have on new editors of Wikipedia. Kittur and Kraut [31] showed that an increase in participation can be achieved by directly delegating specific tasks to contributors. As simple as this approach may appear, the identification of work (and thus specific tasks) is still a tedious and time-consuming process, which can only partly be automated due to its assigned complexity.

With the analysis that we described here, we provide new results that we can use to tackle some of the problems for collaborative authoring systems. These problems are also present in collaborative ontology-engineering projects. For example, we can identify new tasks by combining the results of the *User-Sequence Paths* (Section 4.1) and *Property Paths* (Section 4.3) analyses to suggest classes and the corresponding properties to work on to users.

6.3. Sequential pattern mining

In 1995 Agrawal and Srikant [48] have first addressed the problem of sequential pattern mining. They stated that given a collection of chronologically ordered sequences, sequential pattern mining is about discovering all sequential patterns weighted

according to the number of sequences that contain these patterns. The presented algorithm represents one of the first *a priori* sequential pattern mining algorithms. This means that a specific pattern cannot occur more frequently (above a threshold) if a sub-pattern of this pattern occurs less often (below that threshold). Other examples of *a priori* algorithms are [49, 50].

One of the biggest problems assigned to the *a priori* based sequential pattern mining algorithms was (in the worst case) the exponential number of candidate generation. To tackle this problem Han et al. [51] developed the FP-Growth algorithm.

Many researchers have adapted different algorithms and approaches for different domains to anticipate changing requirements, such as Wang and Han [52] and Hsu et al. [53] who analyzed algorithms for sequential pattern mining in the biomedical domain.

In Walk et al. [30] the authors have presented a novel application of Markov chains to mine and determine sequential patterns from the structured logs of changes of collaborative ontology-engineering projects. Making use of this framework we investigate differences and commonalities across five different collaborative ontology-engineering projects from the biomedical domain.

7. Conclusions & future work

In this work, we discovered intriguing social and sequential patterns that suggest that large collaborative ontology-engineering projects are governed by a few general principles that determine and drive development. Specifically, our results indicate that patterns can be found in all investigated projects, even though the National Cancer Institute Thesaurus (NCIt), the International Classification of Diseases (ICD-11), the International Classification of Traditional Medicine (ICTM), the Ontology for Parasite Lifecycle (OPL) and the Biomedical Resource Ontology (BRO) (i) represent different projects with different goals, (ii) use variations of the same ontology-editors and tools for the engineering process and (iii) differ in the way the projects are coordinated. Using the presented Markov chain analysis, multiple different user-roles could be identified in all investigated datasets. We were also able to see that users work in micro-workflows, meaning that given a specific user, we can identify the most likely users that are editing a specific class next, again independent from the investigated project. When contributing to a project that is created using WebProtégé, iCAT, iCAT-TM or Collaborative Protégé, users exhibit a tendency to do so in a *top-down* and *breadth-first* manner, editing primarily closely related classes while moving along the ontological hierarchy. In ICD-11 and ICTM we were able to identify property-based workflows, meaning that users concentrate their efforts on adding and editing values for one specific property for multiple classes.

The analysis presented not only provides new insights about the engineering and development processes of each single project, but also shows that the analysis of sequential patterns potentially provides actionable insights for different stakeholders in collaborative ontology-engineering projects.

Furthermore, the information of the next possible action (e.g., a user, a change-type, a property, set of classes) or the combination of multiple of these next actions could be used by ontology-engineering tool developers to potentially augment users in collaboratively creating an ontology. For example, by making use of the *Property Paths* analysis to highlight, prefetch, rearrange or adjust sections and content of the interface dynamically, according to the user's needs.

The next logical step to further deepen our understanding of collaborative ontology-engineering projects involves applying the gathered results to productive and live environments, for example as plug-in for (Web)Protégé. Simultaneously, this would allow us to collect valuable data to quantify the usefulness and actionability of the results, generated with our presented approach, in real world scenarios.

Additionally, expanding the Markov chain analysis to take other types of interactions (e.g., discussions, automatic imports and term suggestions by users) into account, represents a potential topic of future work. This also includes a detailed analysis of human factors studies in terms of user-studies (e.g., with a heuristic evaluation or A/B testing) or more sophisticated approaches, such as eye tracking, to assess the usefulness of the presented results for augmenting users when collaboratively engineering an ontology.

Furthermore, as change tracking and click tracking data will likely become available more broadly in the future, we believe that the analysis of this paper and the possible benefits of putting the results into practical use represent an important step towards the development of better (and simpler) ontology editors, which can dynamically anticipate the editing-style of the users. Project administrators could make use of the results of the analysis, for example by allowing for easier delegation of work to the "right" users. This is even more emphasized when considering that the Markov chain analysis is not computationally intensive, making it highly suitable for productive use.

As biomedical ontologies play an increasingly critical role in acquiring, representing, and processing information about human health, we can use quantitative analysis of editing behavior to generate potentially useful insights for building better tools and infrastructures to support these tasks.

Acknowledgement

This work was generously funded by a Marshall Plan Scholarship with support from Graz University of Technology. Further, this work is supported in part by grants GM086587 and GM103316 from U.S. National Institutes of Health.

References

- [1] T. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition* 5 (1993) 199–220.
- [2] W. Borst, Construction of engineering ontologies for knowledge sharing and reuse (1997).
- [3] R. Studer, V. R. Benjamins, D. Fensel, *Knowledge engineering: Principles and methods*, volume 25, 1998, pp. 161–197.
- [4] N. F. Noy, T. Tudorache, Collaborative ontology development on the (semantic) web., in: *AAAI Spring Symposium: Symbiotic Relationships*

- between Semantic Web and Knowledge Engineering, AAAI, 2008, pp. 63–68.
- [5] T. Groza, T. Tudorache, M. Dumontier, Commentary: State of the art and open challenges in community-driven knowledge curation, *Journal of Biomedical Informatics* 46 (2013) 1–4. URL: <http://dx.doi.org/10.1016/j.jbi.2012.11.007>. doi:10.1016/j.jbi.2012.11.007.
 - [6] M. Krötzsch, D. Vrandečić, M. Völkel, Semantic MediaWiki, in: *Proceedings of the 5th International Semantic Web Conference 2006 (ISWC 2006)*, Springer, 2006, pp. 935–942.
 - [7] S. Auer, S. Dietzold, T. Riechert, OntoWiki—A Tool for Social, Semantic Collaboration, in: *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, volume LNCS 4273, Springer, Athens, GA, 2006.
 - [8] C. Ghidini, B. Kump, S. Lindstaedt, N. Mahbub, V. Pammer, M. Rospocher, L. Serafini, MoKi: The Enterprise Modelling Wiki, in: L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, E. P. B. Simperl (Eds.), *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications 2009*, Springer, Berlin, Heidelberg, 2009, pp. 831–835.
 - [9] T. Schandl, A. Blumauer, Poolparty: SKOS thesaurus management utilizing linked data, *The Semantic Web: Research and Applications* 6089 (2010) 421–425.
 - [10] T. Tudorache, C. Nyulas, N. F. Noy, M. A. Musen, WebProtégé: A Distributed Ontology Editor and Knowledge Acquisition Tool for the Web, *Semantic Web Journal* 4 (2013) 89–99.
 - [11] T. Tudorache, S. M. Falconer, C. I. Nyulas, N. F. Noy, M. A. Musen, Will Semantic Web technologies work for the development of ICD-11?, in: *Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*, ISWC (In-Use), Springer, Shanghai, China, 2010.
 - [12] J. Pöschko, M. Strohmaier, T. Tudorache, N. F. Noy, M. A. Musen, Pragmatic analysis of crowd-based knowledge production systems with icat analytics: Visualizing changes to the icd-11 ontology, in: *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium: Wisdom of the Crowd*, Stanford, CA, USA, 2012.
 - [13] S. Walk, J. Pöschko, M. Strohmaier, K. Andrews, T. Tudorache, C. Nyulas, M. A. Musen, N. F. Noy, PragmatiX: An Interactive Tool for Visualizing the Creation Process Behind Collaboratively Engineered Ontologies, *International Journal on Semantic Web and Information Systems* (2013).
 - [14] S. M. Falconer, T. Tudorache, N. F. Noy, An analysis of collaborative patterns in large-scale ontology development projects., in: M. A. Musen, J. Corcho (Eds.), *K-CAP*, ACM, 2011, pp. 25–32.
 - [15] C. Pesquita, F. M. Couto, Predicting the extension of biomedical ontologies, *PLoS Comput Biol* 8 (2012) e1002630. URL: <http://dx.doi.org/10.1371/journal.pcbi.1002630>. doi:10.1371/journal.pcbi.1002630.
 - [16] R. S. Gonçalves, B. Parsia, U. Sattler, Analysing the evolution of the nci thesaurus, in: *Proceedings of the 2011 24th International Symposium on Computer-Based Medical Systems, CBMS '11*, IEEE Computer Society, Washington, DC, USA, 2011, pp. 1–6. URL: <http://dx.doi.org/10.1109/CBMS.2011.5999163>. doi:10.1109/CBMS.2011.5999163.
 - [17] R. S. Gonçalves, B. Parsia, U. Sattler, Facilitating the analysis of ontology differences, in: *Proceedings of the Joint Workshop on Knowledge Evolution and Ontology Dynamics (EvoDyn)*, 2011.
 - [18] R. S. Gonçalves, B. Parsia, U. Sattler, Categorising logical differences between owl ontologies, in: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11*, ACM, New York, NY, USA, 2011, pp. 1541–1546. URL: <http://doi.acm.org/10.1145/2063576.2063797>. doi:10.1145/2063576.2063797.
 - [19] N. F. Noy, A. Chugh, W. Liu, M. A. Musen, A framework for ontology evolution in collaborative environments, in: *The Semantic Web-ISWC 2006*, Springer, 2006, pp. 544–558.
 - [20] B. C. Grau, I. Horrocks, Y. Kazakov, U. Sattler, Just the right amount: extracting modules from ontologies, in: *Proceedings of the 16th international conference on World Wide Web*, ACM, 2007, pp. 717–726.
 - [21] B. C. Grau, I. Horrocks, Y. Kazakov, U. Sattler, A logical framework for modularity of ontologies, in: *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007, pp. 298–303. URL: <http://dl.acm.org/citation.cfm?id=1625275.1625322>.
 - [22] E. Mikroyannidi, L. Iannone, R. Stevens, A. Rector, Inspecting regularities in ontology design using clustering, in: *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I, ISWC'11*, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 438–453. URL: <http://dl.acm.org/citation.cfm?id=2063016.2063045>.
 - [23] M. Strohmaier, S. Walk, J. Pöschko, D. Lamprecht, T. Tudorache, C. Nyulas, M. A. Musen, N. F. Noy, How ontologies are made: Studying the hidden social dynamics behind collaborative ontology engineering projects, *Web Semantics: Science, Services and Agents on the World Wide Web* 20 (2013). URL: <http://www.websemanticsjournal.org/index.php/ps/article/view/333>.
 - [24] H. Wang, T. Tudorache, D. Dou, N. F. Noy, M. A. Musen, Analysis of user editing patterns in ontology development projects, in: *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, Springer, 2013, pp. 470–487.
 - [25] S. Staab, R. Studer, *Handbook on Ontologies*, 2nd ed., Springer Publishing Company, Incorporated, 2009.
 - [26] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, New York, NY, USA, 2003.
 - [27] N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W.-L. Shaiu, L. W. Wright, NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information, *Journal of Biomedical Informatics* 40 (2007) 30–43.
 - [28] J. D. Tenenbaum, P. L. Whetzel, K. Anderson, C. D. Borromeo, I. D. Dinov, D. Gabriel, B. A. Kirschner, B. Mirel, T. D. Morris, N. F. Noy, C. Nyulas, D. Rubenson, P. R. Saxman, H. Singh, N. Whelan, Z. Wright, B. D. Athey, M. J. Becich, G. S. Ginsburg, M. A. Musen, K. A. Smith, A. F. Tarantal, D. L. Rubin, P. Lyster, The Biomedical Resource Ontology (BRO) to enable resource discovery in clinical and translational research, *Journal of Biomedical Informatics* 44 (2011) 137–145.
 - [29] P. Singer, D. Helic, B. Taraghi, M. Strohmaier, Memory and structure in human navigation patterns, *arXiv preprint arXiv:1402.0790* (2014).
 - [30] S. Walk, P. Singer, M. Strohmaier, D. Helic, N. F. Noy, M. A. Musen, Sequential usage patterns in collaborative ontology-engineering projects, *arXiv preprint arXiv:1403.1070* (2014).
 - [31] A. Kittur, R. E. Kraut, Harnessing the wisdom of crowds in wikipedia: quality through coordination, in: *Proceedings of the 2008 ACM conference on Computer supported cooperative work, CSCW '08*, ACM, New York, NY, USA, 2008, pp. 37–46.
 - [32] F. Chierichetti, R. Kumar, P. Raghavan, T. Sarlos, Are web users really markovian?, in: *Proceedings of the 21st international conference on World Wide Web, WWW '12*, ACM, New York, NY, USA, 2012, pp. 609–618. URL: <http://doi.acm.org/10.1145/2187836.2187919>. doi:10.1145/2187836.2187919.
 - [33] J. Borges, M. Levene, Evaluating variable-length markov chain models for analysis of user web navigation sessions, *IEEE Trans. on Knowl. and Data Eng.* 19 (2007) 441–452. URL: <http://dx.doi.org/10.1109/TKDE.2007.1012>. doi:10.1109/TKDE.2007.1012.
 - [34] M. Deshpande, G. Karypis, Selective markov models for predicting web page accesses, *ACM Trans. Internet Technol.* 4 (2004) 163–184. URL: <http://doi.acm.org/10.1145/990301.990304>. doi:10.1145/990301.990304.
 - [35] R. Lempel, S. Moran, The stochastic approach for link-structure analysis (salsa) and the tlc effect, *Comput. Netw.* 33 (2000) 387–401. URL: [http://dx.doi.org/10.1016/S1389-1286\(00\)00034-7](http://dx.doi.org/10.1016/S1389-1286(00)00034-7). doi:10.1016/S1389-1286(00)00034-7.
 - [36] P. L. T. Piorolli, J. E. Pitkow, Distributions of surfers' paths through the world wide web: Empirical characterizations, *World Wide Web* 2 (1999) 29–45. URL: <http://dx.doi.org/10.1023/A:1019288403823>. doi:10.1023/A:1019288403823.
 - [37] R. Sen, M. Hansen, Predicting a web user's next access based on log data, *Journal of Computational Graphics and Statistics* 12 (2003) 143–155. URL: <http://citeseer.ist.psu.edu/sen03predicting.html>.
 - [38] I. Zukerman, D. W. Albrecht, A. E. Nicholson, Predicting users' requests on the www, *Proceedings of the Seventh International Conference on User Modeling*, Springer-Verlag

- New York, Inc., Secaucus, NJ, USA, 1999, pp. 275–284. URL: <http://dl.acm.org/citation.cfm?id=317328.317370>.
- [39] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, in: Proceedings of the seventh international conference on World Wide Web 7, WWW7, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1998, pp. 107–117.
- [40] J. Borges, M. Levene, Data mining of user navigation patterns, in: Revised Papers from the International Workshop on Web Usage Analysis and User Profiling, WEBKDD '99, Springer-Verlag, London, UK, UK, 2000, pp. 92–111. URL: <http://dl.acm.org/citation.cfm?id=648036.744399>.
- [41] I. Cadez, D. Heckerman, C. Meek, P. Smyth, S. White, Model-based clustering and visualization of navigation patterns on a web site, *Data Min. Knowl. Discov.* 7 (2003) 399–424. URL: <http://dx.doi.org/10.1023/A:1024992613384>. doi:10.1023/A:1024992613384.
- [42] R. R. Sarukkai, Link prediction and path analysis using markov chains, Proceedings of the 9th international World Wide Web conference on Computer networks: the international journal of computer and telecommunications network, North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 2000, pp. 377–386. URL: <http://dl.acm.org/citation.cfm?id=347319.346322>.
- [43] A. Cabrera, E. F. Cabrera, Knowledge-Sharing Dilemmas, *Organization Studies* 23 (2002) 687–710.
- [44] B. Keegan, D. Gergle, N. S. Contractor, Hot off the wiki: dynamics, practices, and structures in Wikipedia's coverage of the Tohoku catastrophes., in: F. Ortega, A. Forte (Eds.), *Int. Sym. Wikis*, ACM, 2011, pp. 105–113.
- [45] N. Shachaf, Beyond vandalism: Wikipedia trolls., *Journal of Information Science*; Jun2010, Vol. 36 Issue 3, p357-370, 14p, 2 Charts (2010).
- [46] B. Suh, G. Convertino, E. H. Chi, P. Pirolli, The singularity is not near: slowing growth of wikipedia, in: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, ACM, New York, NY, USA, 2009, pp. 1–10.
- [47] A. Halfaker, A. Kittur, J. Riedl, Don't bite the newbies: how reverts affect the quantity and quality of Wikipedia work., in: F. Ortega, A. Forte (Eds.), *Int. Sym. Wikis*, ACM, 2011, pp. 163–172.
- [48] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proceedings of the Eleventh International Conference on Data Engineering, ICDE '95, IEEE Computer Society, Washington, DC, USA, 1995, pp. 3–14. URL: <http://dl.acm.org/citation.cfm?id=645480.655281>.
- [49] R. T. Ng, L. V. S. Lakshmanan, J. Han, A. Pang, Exploratory mining and pruning optimizations of constrained associations rules, in: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD '98, ACM, New York, NY, USA, 1998, pp. 13–24. URL: <http://doi.acm.org/10.1145/276304.276307>. doi:10.1145/276304.276307.
- [50] S. Sarawagi, S. Thomas, R. Agrawal, Integrating association rule mining with relational database systems: Alternatives and implications, in: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD '98, ACM, New York, NY, USA, 1998, pp. 343–354. URL: <http://doi.acm.org/10.1145/276304.276335>. doi:10.1145/276304.276335.
- [51] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, ACM, New York, NY, USA, 2000, pp. 1–12. URL: <http://doi.acm.org/10.1145/342009.335372>. doi:10.1145/342009.335372.
- [52] J. Wang, J. Han, Bide: Efficient mining of frequent closed sequences, in: Proceedings of the 20th International Conference on Data Engineering, ICDE '04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 79–. URL: <http://dl.acm.org/citation.cfm?id=977401.978142>.
- [53] C.-M. Hsu, C.-Y. Chen, B.-J. Liu, C.-C. Huang, M.-H. Laio, C.-C. Lin, T.-L. Wu, Identification of hot regions in protein-protein interactions by sequential pattern mining, *BMC bioinformatics* 8 (2007) S8.